UIUCDCS-R-79-966

UILU-ENG 79 1712

THE MINIMAL COVERING PROBLEM AND AUTOMATED DESIGN OF TWO-LEVEL AND/OR OPTIMAL NETWORKS

by

MING HUEI YOUNG

March 1979

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

Report No. UIUCDCS-R-79-966

THE MINIMAL COVERING PROBLEM
AND AUTOMATED DESIGN OF
TWO-LEVEL AND/OR OPTIMAL NETWORKS

by

MING HUEI YOUNG

March 1979

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

This work was supported in part by the National Science Foundation under Grant No. MCS77-09744 and was submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science, March 1979.

ACKNOWLEDGMENT

The author wishes to thank his advisor, Professor S. Muroga, for his invaluable guidance during the preparation of this thesis and during the preceding years of research, and also for his careful reading and constructive criticism of the original manyscript.

The author also wishes to thank Professor J. Liebman, who gave careful proof reading and helpful criticism of this thesis, and to Mr. R. B. Cutler, who gave helpful comments and provided friendship during the author's study here.

The excellent typing job done by Mrs. Ruby Taylor, Mrs. Zigrida Arbatsky, and Miss Kim Howard is appreciated and acknowledged.

The financial support of the Department of Computer Science and the National Science Foundation under Grant No. MCS77-09744 is acknowledged.

Special thanks goes to Mrs. Ming-Chiao Lien Young for her patience, understanding, and encouragement during the author's five years here.

TABLE OF CONTENTS

Page

THE MINIMAL COVERING PROBLEM

AND

AUTOMATED DESIGN OF TWO-LEVEL AND/OR OPTIMAL NETWORKS

Ming Huei Young
Department of Computer Science
University of Illinois at Urbana-Champaign, 1978

Efficient implicit enumeration algorithms for the minimal covering problem are presented in this thesis. These algorithms are developed mainly for minimizing the logic expression of the switching function. They are extensions of the Quine-McCluskey method.

"The reducing property" and "the excluding property" of the minimal covering problem are introduced to speed up the enumeration in solving problems.

Symmetric property of the minimal covering problem is extensively explored. Procedures for utilizing this property in the implicit enumeration algorithm are developed based on the theory of finite permutation group.

The concept of an upper bound on the value of a group and of variable is also introduced in this thesis.

Programs developed based on these algorithms are incorporated into a system for the automated design of two-level AND/OR optimal networks.

1. INTRODUCTION

The logic minimization problem is an important logic design problem. This problem is to find a minimal set of terms for a switching function* such that this function can be expressed as a sum or sums of these products. A switching function expressed in a disjunction or disjunctions of terms can be easily implemented with PLAs (Programmable Logic Arrays). Since the size of a PLA for implementing a switching function in a disjunction of terms or disjunctions of terms is proportional to the number of different terms in this disjunction or these disjunctions, minimization of the logic expression for a switching function means minimization of the size of a PLA.

In implementing a PLA as part of LSI chips, chip areas are covered by the PLA or electric power consumed by the PLA is minimized if the logic expression of a function to be implemented is minimized. In using each PLA as a separate package, minimization of the logic expression of each switching function usually reduces the number of packages needed to implement these functions, if a large network is to be realized by many packages.

The most well-known method for the logic minimization problem is the Quine-McCluskey method [6]. This method consists of

---

* A switching function may be a single-output or a multiple-output switching function unless it is explicitly specified.

two stages. The first stage is to derive the set of all potential terms for the switching function. The second stage is to find a minimal set of terms from the set derived in the first stage. In this method, the problem of the second stage is formulated as a minimal covering problem and is solved by a "reduction and branching" method. Three reduction operations are used in this method to reduce a problem into a smaller equivalent problem. When a problem cannot further be reduced, this problem is decomposed into subproblems by fixing some variable to 0 and 1, and each subproblem is solved individually by repeating the reduction and branching.

In this thesis, a zero-one implicit enumeration algorithm for the minimal covering problem is introduced. This algorithm is an extension of the Quine-McCluskey method. Some new properties of the minimal covering problem, which can be used to speed up the Quine-McCluskey method, are incorporated in this algorithm. These properties are presented in Chapters 5, 7, 8 and 9. An heuristic algorithm for large-scale minimal covering problems is proposed in Chapter 6.

If the given switching function has some symmetric properties, these properties are reflected in the minimal covering problem formulated for the minimization of the logic expression of this function. These are also discussed in Chapter 7.

Some new properties presented in Chapters 5 and 7 are generalized for the general cost minimal covering problems in Chapter 10.

Although this algorithm is developed mainly for the minimization problem of logic expressions, it can also be applied to minimal covering problems or general cost minimal covering problems formulated for other problems [1, 2, 3, 4, 5]. For example, problems [24],

which the late Professor Fulkerson of Cornell concluded were difficult to solve, were solved by the program based on this algorithm. Comparison of computational results shows that this algorithm is one of the best algorithms for the minimal covering problem.

2. FORMULATION OF THE LOGIC MINIMIZATION PROBLEM INTO THE MINIMAL COVERING PROBLEM

A method for formulating the logic minimization problem into the minimal covering problem is described in this chapter. This is the method described in [6].

2.1 Single-output And Multiple-output Switching Functions

Let B be a set with only two elements 0 and 1. Three logic operations AND, OR, and NOT are denoted by ".", "$\vee$", and "-", respectively.

Let B^t be the set of all t-vectors $(y_1, y_2, \ldots, y_t)$ such that $y_i = 0$ or 1 for $i=1, 2, \ldots, t$. A single-output switching function $f(y_1, y_2, \ldots, y_t)$ on B^t is a mapping from B^t to B. Each single-output switching function f on B^t can be expressed [6, 30] as a disjunction of terms:

$$f(y_1, y_2, \ldots, y_t) = Z_{i_1} \cdot Z_{i_2} \cdot \ldots \cdot Z_{i_\alpha} \vee Z_{k_1} \cdot Z_{k_2} \cdot \ldots \cdot Z_{k_\beta} \vee \ldots \vee Z_{\ell_1} \cdot Z_{\ell_2} \cdot \ldots \cdot Z_{\ell_\gamma} \qquad (2.1.1)$$

where $Z_r = y_{s(r)}$ or $\bar{y}_{s(r)}$ for some $s(r)$ in $\{1, 2, \ldots, t\}$ for each $r = i_1, i_2, \ldots, i_\alpha, k_1, k_2, \ldots, k_\beta, \ldots, \ell_1, \ell_2, \ldots, \ell_\gamma$, and each Z_r is called a literal.

A multiple-output switching function is a set of single-output switching functions defined on B^t.

2.2 Logic Minimization For A Single-output Switching Function

Let $f(y_1, y_2, \ldots, y_t)$ and $g(y_1, y_2, \ldots, y_t)$ be two single-output switching functions. If every $(y_1, y_2, \ldots, y_t)$

satisfying $f(y_1, y_2, \ldots, y_t) = 1$ satisfies also $g(y_1, y_2, \ldots, y_t) = 1$, then f is said to *imply* g. For example, $f(y_1, y_2, y_3) = y_1 \cdot y_2 \vee y_2 \cdot \bar{y}_3$ implies $g(y_1, y_2, y_3) = y_1 \cdot y_2 \vee y_2 \cdot \bar{y}_3 \vee y_1 \cdot y_3$. An *implicant* of a single-output switching function f is a product which implies f. For example, $y_i \cdot y_2$ is an implicant of $f(y_1, y_2, y_3) = y_1 \cdot y_2 \vee y_2 \cdot \bar{y}_3$. A product $p = Z_{i_1} \cdot Z_{i_2} \cdot \ldots \cdot Z_{i_{j(i)}}$ is said to *subsume* another product $q = Z_{k_1} \cdot Z_{k_2} \cdot \ldots \cdot Z_{k_{j(k)}}$ if each literal (i.e., Z_{k_i}) of q is a literal of p. For example, the product $y_1 \cdot y_2 \cdot \bar{y}_3$ subsumes the product $y_2 \cdot \bar{y}_3$. A *prime implicant* of a single-output switching function f is defined as an implicant of f such that no other product subsumed by it can be an implicant of f. For example, $y_1 \cdot y_2$, $\bar{y} \cdot y_3$ and $y_1 \cdot y_3$ are prime implicants of $f(y_1, y_2, y_3) = y_1 \cdot y_2 \vee \bar{y}_2 \cdot y_3$. A vector $(y_1, y_2, \ldots, y_t)$ is said to be a *true vector* of a single-output switching function f if $f(y_1, y_2, \ldots, y_t) = 1$. For example, (1, 1, 0) is a true vector of the function $f(y_1, y_2, y_3) = y_1 \cdot y_2 \vee y_2 \cdot \bar{y}_3$. A true vector of a single-output switching function f is said to be *covered* by the prime implicant q_i of f if this true vector is also a true vector of q_i. Each prime implicant of the single-output switching function f covers some true vectors of f. If all the true vectors of the single-output switching function f are covered by prime implicants $q_{k_1}, \ldots, q_{k_r}$, then $f = q_{k_1} \vee q_{k_2} \vee \ldots \vee q_{k_1}$ holds and $\{q_{k_1}, q_{k_2}, \ldots, q_{k_r}\}$ is called a *realization set* of f.

To find a minimal set of terms to express a single-output

switching function f as a disjunction of these terms, all the prime implicants of f are first found, and then a minimal set of prime implicants are chosen from these prime inplicants such that all the true vectors of f are covered by those prime implicants in it. All the prime implicants of a given function f can be found by a method called <u>iterated consensus</u>, or some other methods [30, 31]. Let $q_1, q_2, \ldots, q_n$ be all the prime implicants of a single-output switching function f, and $\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_m$ be all the true vectors of f. Let $A = [a_{ij}]$ be an $m \times n$ matrix, where a_{ij} is defined as

$$a_{ij} = \begin{cases} 1 & \text{if } \vec{y}_i \text{ is covered by } q_j, \\ 0 & \text{if } \vec{y}_i \text{ is not covered by } q_j, \end{cases}$$

for all i, j. The matrix A is called the <u>prime implicant table</u>* of the single-output of the switching function f. For each i, let x_i be a zero-one variable such that if $x_i = 1$, prime implicant q_i is to be chosen in a realization solution set for f, and if $x_i = 0$, q_i is not. Then the logic minimization problem of a single-output switching function can be formulated as the following minimal covering problem:

$$\text{minimize: } x_i + x_2 + \ldots + x_n$$

---

* The prime implicant table defined here is different from that defined in textbooks of switching theory in that rows and columns are interchanged.

subject to:

$$A \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

$$x_i = 0 \text{ or } 1 \text{ for } i = 1, 2, \ldots, n,$$

where A is the prime implicant table of f.

<u>Example 2.2.1</u> Let us consider the minimization problem of the switching function

$$f(y_1, y_2, y_3, y_4) = y_1 \cdot y_2 \cdot \bar{y}_3 \vee y_2 \cdot y_3 \cdot \bar{y}_4 \vee \bar{y}_1 \cdot y_3 \cdot y_4 \vee y_1 \cdot \bar{y}_2 \cdot y_4 .$$

All the prime implicants found by the iterated consensus method are: $q_1 = y_1 \cdot y_2 \cdot \bar{y}_3$, $q_2 = y_2 \cdot y_3 \cdot \bar{y}_4$, $q_3 = \bar{y}_1 \cdot y_3 \cdot y_4$, $q_4 = y_1 \cdot \bar{y}_2 \cdot y_4$, $q_5 = y_1 \cdot y_2 \cdot \bar{y}_4$, $q_6 = \bar{y}_1 \cdot y_2 \cdot y_3$, $q_7 = y_1 \cdot \bar{y}_3 \cdot y_4$, $q_8 = \bar{y}_2 \cdot y_3 \cdot y_4$. All the true vectors of this function are: $\vec{y}_1 = (1, 1, 0, 0)$, $\vec{y}_2 = (0, 1, 1, 0)$, $\vec{y}_3 = (1, 1, 1, 0)$, $\vec{y}_4 = (1, 0, 0, 1)$, $\vec{y}_5 = (1, 1, 0, 1)$, $\vec{y}_6 = (0, 0, 1, 1)$, $\vec{y}_7 = (1, 0, 1, 1)$, $\vec{y}_8 = (0, 1, 1, 1)$. The prime implicant table of this function is as follows:

$$
A = \begin{array}{c} \\ \vec{y}_1 \\ \vec{y}_2 \\ \vec{y}_3 \\ \vec{y}_4 \\ \vec{y}_5 \\ \vec{y}_6 \\ \vec{y}_7 \\ \vec{y}_8 \end{array}
\begin{array}{c}
\begin{array}{cccccccc} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 \end{array} \\
\left(\begin{array}{c|c|c|c|c|c|c|c}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ \hline
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ \hline
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0
\end{array}\right)
\end{array} . \qquad (2.2.1)
$$

So the minimal covering problem formulated for this problem is to

minimize $x_1 + x_2 + x_3 + x_4 + x_5 + x_7 + x_8$,

subject to

$$
A \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} ,
$$

$$x_i = 0 \text{ or } 1, \text{ for } i = 1, 2, \ldots, 8,$$

where A is the matrix of (2.2.1). $\blacksquare$

2.3 Logic Minimization For a Multiple-output Switching Function

The problem to find a minimal set of terms for multiple-output switching function $\{f_1, f_2, \ldots, f_\mu\}$ such that each single output

switching function f_i can be expressed as a disjunction of some terms in this set is more complicated than the problem for the single-output switching function.

All possible products of $f_1, f_2, \ldots, f_\mu$ are first formed, such as $f_1 \cdot f_2, \ldots,$ and $f_1 \cdot f_2 \cdot \ldots \cdot f_\mu$. Let $\Psi_1, \Psi_2, \ldots, \Psi_\ell$ denote $f_1, f_2, \ldots, f_\mu$ and all their products. Then all prime implicants for each Ψ_i and true vectors for each f_j are derived. Let $q_{i_1}, q_{i_2}, \ldots, q_{i_{n(i)}}$ be all the prime implicants of Ψ_i for each $i = 1, 2, \ldots, \ell$, and let $\vec{y}_{j_1}, \vec{y}_{j_2}, \ldots, \vec{y}_{j_{m(j)}}$ be all the true vectors of f_j for each $j = 1, 2, \ldots, \mu$. Then an $m \times n$ matrix which indicates which truth vectors are covered by each q_{i_k} is constructed, where $m = m(1) + \ldots + m(j) + \ldots + m(\mu)$ and $n = n(1) + \ldots + n(i) + \ldots + n(\ell)$. In constructing this matrix, each prime implicant q_{i_k} of Ψ_i covers only the true vectors of the single-output functions contained in the product Ψ_i. Then a minimal covering problem is formulated from this $m \times n$ matrix in the same manner as in the single-output switching function case.

Example 2.2.2 Let us consider the logic minimization problem of the multiple-output function $\{f_1 = y_2 \cdot y_4 \vee y_1 \cdot y_4 \vee \bar{y}_1 \cdot \bar{y}_3 \cdot \bar{y}_4,$ $f_2 = y_1 \cdot \bar{y}_3 \vee \bar{y}_1 \cdot y_2 \cdot y_4 \vee \bar{y}_1 \cdot \bar{y}_2 \cdot y_4\}$. The product of f_1 and f_2 is $f_1 \cdot f_2 = \bar{y}_1 \cdot y_2 \cdot y_4 \vee y_1 \cdot \bar{y}_3 \cdot y_4 \vee \bar{y}_1 \cdot \bar{y}_2 \cdot \bar{y}_3 \cdot \bar{y}_4$. All prime implicants of f_1, f_2 and $f_1 \cdot f_2$ are : $q_1 = y_2 \cdot y_4$, $q_2 = y_1 \cdot y_4$, $q_3 = \bar{y}_1 \cdot \bar{y}_3 \cdot \bar{y}_4$, $q_4 = y_1 \cdot \bar{y}_2 \cdot \bar{y}_3$, $q_5 = y_1 \cdot \bar{y}_3$, $q_6 = \bar{y}_1 \cdot y_2 \cdot y_4$, $q_7 = \bar{y}_1 \cdot \bar{y}_2 \cdot \bar{y}_3 \cdot \bar{y}_4$, $q_8 = y_2 \cdot \bar{y}_3 \cdot y_4$, $q_9 = \bar{y}_1 \cdot y_2 \cdot y_4$, $q_{10} = y_1 \cdot \bar{y}_3 \cdot y_4$, $q_{11} = \bar{y}_1 \cdot \bar{y}_2 \cdot \bar{y}_3 \cdot \bar{y}_4$, and $q_{12} = y_2 \cdot \bar{y}_3 \cdot y_4$.

The true vectors of f_1 are: $\vec{y}_1 = (0, 0, 0, 0)$, $\vec{y}_2 = (0, 1, 0, 0, 0)$ $\vec{y}_3 = (0, 1, 0, 1)$, $\vec{y}_4 = (0, 1, 1, 1)$, $\vec{y}_5 = (1, 0, 0, 1)$, $\vec{y}_6 = (1, 0, 1, 1)$, $\vec{y}_7 = (1, 1, 0, 1)$, $\vec{y}_8 = (1, 1, 1, 1)$. The true vectors of f_2 are: $\vec{y}_9 = (0, 0, 0, 0)$, $\vec{y}_{10} = (0, 0, 1, 0)$, $\vec{y}_{11} = (0, 1, 0, 1)$, $\vec{y}_{12} = (0, 1, 1, 1)$, $\vec{y}_{13} = (1, 0, 0, 0)$, $\vec{y}_{14} = (1, 0, 0, 1)$, $\vec{y}_{15} = (1, 1, 0, 0)$ $\vec{y}_{16} = (1, 1, 0, 1)$.

The prime implicant table for this multiple-output function is:

		f_1				f_2				$f_1 \cdot f_2$			
		q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}
	$\vec{y}_1$	0	0	1	0					0	0	0	0
	$\vec{y}_2$	0	0	1	1					0	0	0	0
	$\vec{y}_3$	1	0	0	1		0			1	0	1	1
f_1	$\vec{y}_4$	1	0	0	0					1	0	0	0
	$\vec{y}_5$	0	1	0	0					0	1	0	0
	$\vec{y}_6$	0	1	0	0					0	0	0	0
	$\vec{y}_7$	1	1	0	0		0			0	1	1	1
	$\vec{y}_8$	1	1	0	0					0	0	0	0
	$\vec{y}_9$					0	0	1	0	0	0	0	0
	$\vec{y}_{10}$					0	0	1	0	0	0	0	0
	$\vec{y}_{11}$		0			0	1	0	1	1	0	1	1
f_2	$\vec{y}_{12}$					0	1	0	0	1	0	0	0
	$\vec{y}_{13}$					1	0	0	0	0	0	0	0
	$\vec{y}_{14}$					1	0	0	0	0	1	0	0
	$\vec{y}_{15}$		0			1	0	0	0	0	0	0	0
	$\vec{y}_{16}$					1	0	0	1	0	1	1	1

,

where the area with a single 0 shows that all elements in that area are all zeroes.

The following terminology will be used later.

The set of all multiple-output prime implicants (abbreviated as MOPI) of a multiple-output switching function f is defined as the set of all prime implicants of all possible products Ψ_1, Ψ_2, ..., Ψ_ℓ, of the output functions f_1, f_2, ..., f_μ of f. A multiple-output implicant (abbreviated as MOI) of a multiple-output switching function f is defined as an implicant of some possible product Ψ_i of the output functions f_1, f_2, ..., f_μ of f. A realization set of a multiple-output switching function f is defined as a set of terms such that each output function f_i of f can be expressed as a disjunction of some terms in this set.

3. ZERO-ONE IMPLICIT ENUMERATION ALGORITHM FOR THE MINIMAL COVERING PROBLEM

The zero-one implicit enumeration algorithm for a zero-one integer linear programming problem is first introduced by E. Balas [9]. The basic idea for this algorithm consists of the following steps:

B1. Examine if a subproblem (initially the given problem) can be easily solved or not. If it can be concluded by some means that no solution better than the best solution found so far can be obtained for the current subproblem, go to step B3. If a best solution of the current subproblem is found and if this solution is better than the best solution obtained so far, store this new solution as the best solution obtained so far and go to step B3.

B2. Choose an unfixed variable, which is called a branching variable, and generate two subproblems by fixing the chosen variable to 0 and 1. Store these two subproblems.

B3. Pick one subproblem from the storage where the subproblems are stored and go to B1. If there is no subproblem left in the storage, then the given problem is implicitly enumerated and the best solution obtained so far is an optimal solution for the given problem. ■

In using this implicit enumeration procedure, one must have some easy means to detect that no solution better than the best solution obtained so far can be found for each subproblem. Also, one must have a good way to store all subproblems generated at step B2. The criterion used in B2 for choosing a branching variable has a strong influence on the execution time of the above procedure.

To show how this zero-one implicit enumeration algorithm is applied to the minimal covering problem, some basic concepts are introduced in Section 3.1, and the three reduction operations of the minimal covering problem are restated in Sections 3.2. Then the basic implicit enumeration algorithm for the minimal covering problem is outlined in Section 3.3.

3.1 Some Basic Definitions

A minimal covering problem is a problem to minimize $x_1 + x_2 + \dots + x_n$, subject to:

$$\text{(P)} \qquad A \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ \vdots \\ \end{pmatrix} .$$

$$x_i = 0 \text{ or } 1 \text{ for } i = 1, 2, \dots n,$$

where $A = (a_{ij})$ is an m by n matrix with $a_{ij} = 0$ or 1.

The i-th row of A is said to be covered by the j-th column of A, or the j-th column of A is said to be covered by the i-th row of A if $a_{ij} = 1$.

A solution of the minimal covering problem is defined as an n-vector $(x_1\ x_2, \ldots, x_n)$ with $x_i = 0$ or 1 for $i = 1, 2, \ldots, n$. A solution $(x_1, x_2, \ldots, x_n)$ is said to be a feasible solution of (P) if it satisfies

$$A \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} .$$

An optimal solution of the minimal covering problem (P) is a feasible solution which minimizes $x_1 + x_2 + \ldots + x_n$.

A variable of the problem (P) is said to be fixed if it has been assigned a fixed value 0 or 1. A variable is said to be free if it is not fixed yet. A subproblem is a problem obtained from another problem by fixing some free variables of that problem to 0 or 1. A partial solution of a subproblem is the set of fixed variables of that subproblem. The given problem is considered as a subproblem with an emtpy partial solution. A completion of a partial solution S is defined as a solution that is derived from S by specifying all free variables to 0 or 1. A constraint is said to be satisfied by a partial solution S if it can be satisfied by a completion derived from S by specifying all free variables to 0.

Henceforth, $\vec{a}_i$ and $\vec{r}_j$ denote the i-th column and the j-th row of the matrix A, respectively. It is assumed that each column $\vec{a}_i$ of matrix A contains at least one non-zero element.

3.2 Reduction Operations

The following three reduction operations* are discussed in [6] for reducing a prime implicant table or equivalently, for reducing the constraint matrix of a minimal covering problem.

A column $\vec{a}_j = (a_{ij}, a_{2j}, \ldots, a_{mj})$ (or row $\vec{r}_i$) is said to be dominated by another column $\vec{a}_k = (a_{1k}, a_{2k}, \ldots, a_{mk})$ (or another row $\vec{r}_k$) if $a_{ij} \leq a_{ik}$ for $i = 1, 2, \ldots, m$ (or $a_{ij} \leq a_{kj}$ for $j = 1, 2, \ldots, n$).

Operation 1. If row $\vec{r}_i$ is dominated by row $\vec{r}_j$, then $\vec{r}_j$ is deleted from the constraint matrix.

Operation 2. If column $\vec{a}_j$ is dominated by column $\vec{a}_i$, then column $\vec{a}_j$ is deleted from the matrix and the variable x_j is fixed to 0.

Operation 3. If row $\vec{a}_i$ consists of components $a_{ik} = 1$ for only one k and $a_{ij} = 0$ for all $j \neq k$, then the variable x_k is fixed to 1 and column $\vec{a}_k$ is deleted from the matrix. Also, all rows with k-th component equal to 1 are deleted from the matrix. Column $\vec{a}_k$ is said to be essential.

It is shown in [14] that these three operations can be applied in any order to obtain a unique matrix where none of these operations can be further applied.

---

* Rows and Columns in this thesis are interchanged unlike those in [6].

3.3 Basic Implicit Enumeration Algorithm For The Minimal Covering Problem

In the following outline of the algorithm, each subproblem is stored with a level number, denoted by LEVEL, indicating which subproblem this subproblem is obtained from. At the beginning, the given problem is assigned level number 1. The level number of a subproblem obtained from a subproblem of level k is k + 1. A level number k_1 is said to be higher than another level number k_2 if $k_1 < k_2$. The partial solution of each subproblem is stored in a stack XSL with its level number.

The algorithm is outlined as follows:

M1 Reduction.

Using the operations introduced in 3.2, reduce the constraint matrix as much as possible. Update the current partial solution. If the matrix is reduced to a null matrix go to M5.

M2 Bounding.

M2.1 Find a lower bound ZMIN of the problem under the current partial solution.

M2.2 Test if "ZBAR-ZMIN $\leq$ 0" is satisfied, where ZBAR is the best value obtained so far. If it is satisfied go to M6.

M3 Branching.

M3.1 Choose a row $\vec{r}_i$ by some criterion.

M3.2 Based on the row chosen at M3.1, for each non-zero element a_{ij} in this row, generate a subproblem by fixing variable x_j to 1.

M3.3 Store indices (j,-k) for all subproblems just generated in a stack XX, where j is the index of the branching variable and k is the level number of the subproblem. The problem corresponding to column j with the fewest non-zero elements is stored first.

M4 Next Subproblem.

Get an index (j,-k) from the top of stack XX and set variable x_j to 1. Then delete all rows with the j-th element equal to 1. Update the current partial solution and go to M1.

M5 Getting a feasible solution.

The current partial solution is a feasible solution. If this feasible solution is better than the best feasible solution obtained so far, keep this solution as the best feasible solution found so far.

M6 Backtracking.

M6.1 Find a partial solution, in XSL, one level higher than the current subproblem and consider it as the current partial solution. In XSL, erase all partial solutions with level number greater than the level number of the current partial solution. If no partial solution is left in XSL, the given problem has been implicitly enumerated and the best solution obtained so far is an optimal solution.

M6.2 Retrieve the lower bound ZMIN (which was calculated at M2.1 or M6.6) of the current subproblem.

M6.3 Test if "ZBAR-ZMIN $\leq$ 0" is satisfied. If it is satisfied go to M6.1.

M6.4 Compare the level number LEVEL of the current partial solution with the level number LVT of the next subproblem to be considered in XX. If LVT < LEVEL + 1 go to M6.1. If LVT $\geq$ LEVEL + 1, delete the next subproblem in XX and repeat M6.4.

M6.5 Set the variable, corresponding to the subproblem which has just been implicitly enumerated, to 0 and delete its corresponding column in the constraint matrix.

M6.6 Calculate a lower bound ZMIN of the current subproblem and test if "ZBAR-ZMIN $\leq$ 0" is satisfied. If it is satisfied, go to M6.1. Otherwise, go to M4.

Henceforth, "program backtracks" means that the program control goes to M6 (i.e., a subproblem has been implicitly enumerated and program tries to derive another subproblem).

The method used in M2.2 or M6.6 for finding the lower bound ZMIN of a subproblem with a certain partial solution is the one introduced in [11] and is restated as follows.

Let ℓ_i be the weight of column i (i.e., the number of nonzero elements in column i). Arrange these numbers in a descending order: $\ell_{i_1} \geq \ell_{i_2} \geq \cdots \geq \ell_{i_m}$. Let h be the number of unsatisfied constraints by the current partial solution and r be the smallest integer such that $\ell_{i_1} + \ell_{i_2} + \cdots + \ell_{i_r} \geq h$. Then ZMIN is calculated by ZMIN = r + XP, where XP is the number of variables which are fixed to 1 in the current partial solution.

The criterion used in M3.1 for choosing a row is described as follows:

<u>M3.1.1</u> For each column i calculate n_i = (The number of non-zero elements in column i) + (The number "bicolumnar rows" covered by column i). Here a row is said to be "bicolumnar" if it contains only two non-zero elements.

<u>M3.1.2</u> Find the column i_o with the largest n_{i_o}. If there is a tie, the column with the greatest index is chosen.

<u>M3.1.3</u> From all the rows covered by column i_o, choose the row with the smallest number of non-zero elements. If there is a tie, the row with the smallest index is chosen.

4. SCHEME FOR THE PROBLEM REDUCTION

In using the implicit enumeration algorithm introduced in the last chapter to solve the minimal covering problem, long computation time will have to be spent in the problem size reduction if the column domination relation or the row domination relation are checked for each pair of columns or each pair of rows each time the algorithm goes through the MI Reduction in the previous chapter. The computational efficiency of this algorithm is greatly improved by the use of the scheme introduced in Section 4.1 for checking the domination relations among rows and among columns.

4.1 A Scheme For Detection Of Domination Relations

In the Ml Reduction, the column domination relation and the row domination relation are checked only in the beginning. After a column or a row has been checked not to be dominated by any others, it needs to be checked again only when the existing non-zero elements in it are deleted. So in the Ml Reduction, two arrays, MM1 and MM2, are used to keep track of which columns and which rows need to be checked again.

A column is tested to see if it is dominated by any other columns as follows:

<u>Ml.1</u> Find the first row which is covered by the column to be tested.

<u>Ml.2</u> All the columns that are covered by the row found at step Ml.1 are the candidate columns that may dominate the column to be tested. Check if any of these

columns dominate the column to be tested. ■

A row can be similarly tested.

4.2 Comparison Of Some Computational Results

Comparison of some computational results for the two cases -- with and without the use of the new scheme introduced in Section 4.1 -- has been made on some example problems, as summarized in Table 4.2.1. Programs for these two different cases are coded in FORTRAN and compiled by the FORTRAN G compiler. Computational results are obtained by solving the problems on the IBM 360/75J computer.

PROB. NO.	PROBLEM SIZE		USING CONVENTIONAL PROCEDURE			USING PROCEDURE STATED IN SECTION 4.1		
	m	n	NO. OF ITER	NO. OF BKTRK	TIME IN SEC	NO. OF ITER	NO. OF BKTRK	TIME IN SEC
1	55	44	51*	25	8.97	49	24	1.35
2	35	15	174*	85	5.12	159	77	1.76
3	40	60	5	2	0.72	5	2	0.15
4	60	60	47	24	10.91	47	24	1.17
5	60	80	>250	?	>72.20	350	191	11.60

Table 4.2.1

Comparison of two cases: program with conventional procedure of checking dominating relations and program with the procedure stated in Section 4.1.

* The branching operation is slightly changed due to different checking procedure.

The number in the column under "NO. OF ITER" shows the number of iterations, i.e., the number of times the program went through step M4 in solving a problem. The number in the column under "NO. OF BKTRK" shows the number of backtracks, i.e., the number of times the program went through step M6. The number in the column under "TIME IN SEC" shows the computation time (in seconds) used in solving a problem.

From this table, one can see a great computational improvement due to this new scheme of checking domination relations among rows and columns in spite of the simplicity of the scheme. For problems with greater size, the improvements will be even greater.

5. NEW PROPERTIES OF THE MINIMAL COVERING PROBLEM

Two new properties of the minimal covering problem that can be used to improve the algorithm's efficiency are presented in this chapter.

5.1 Reducibility Of A Partial Solution

A feasible solution $(x_1, x_2, \ldots, x_n)$ of the minimal covering problem (P) is said to be reducible if there exists $x_i = 1$ for some i and $(x_1, x_2, \ldots, x_{j-i}, 0, x_{j+1}, \ldots, x_n)$ is also a feasible solution of (P). Otherwise, a feasible solution is irreducible. It is easy to see the following theorem:

Theorem 5.1.1 Every optimal solution of the minimal covering problem (P) is irreducible.

A partial solution S of a subproblem is reducible if among all the constraints of (P), those which S can satisfy (along with all free variables assigned to 0) can also be satisfied by another partial solution S', which is obtained from S by setting one non-zero variable in S to 0 and keeping all other fixed variables in S unchanged (along with all free variables assigned to 0).

Theorem 5.1.2 Every feasible completion of a reducible partial solution S is reducible.

Proof Let S^* be a feasible completion of S. Since S is reducible, there exists another partial solution S_1 such that

(1) S_1 is obtained from S by setting some fixed variable x_i in S from 1 to 0.

(2) S_1 satisfies all constraints which S can satisfy.

Let S_1^* be a solution obtained from S^* by setting the variable x_i from 1 to 0. We shall show that S_1^* is a feasible solution of (P).

Among all constraints of (P), those which are satisfied by S are satisfied by S_1. Therefore, they are satisfied by S_1^*, since S_1^* is a completion of S_1. (Notice that every coefficient a_{ij} is 0 or 1.)

Now let us consider the constraints which are not satisfied by S but are satisfied by S^*. Let an arbitrary constraint among them be

$$a_{k1}\, x_1 + \ldots + a_{kt}\, x_t + (a_{k,t+1}\, x_{t+1} + \ldots + a_{kn}\, x_n) \geq 1,$$

where $x_{t+1}, \ldots, x_{t+n}$ are the fixed variables (0 or 1) in S. Since this is not satisfied by S but is satisfied by S^*, $a_{k1}\, x_1 + \ldots + a_{kt}\, x_t \geq 1$ must hold for the feasible completion S^* of S. This is true even if x_i is changed from 1 to 0 since x_i is not one of $x_1, \ldots, x_t$. Thus, the constraint which is not satisfied by S but is satisfied by S^* is also satisfied by S_1^*.

Therefore, S^* is reducible.

Q.E.D.

From this theorem, it is easy to see the following corollary.

Corollary 5.1.3 Every completion of a reducible partial solution is either infeasible or not optimal.

Thus, the computational efficiency of the zero-one implicit enumeration algorithm in Chapter 3 can be improved by checking the

reducibility of a partial solution. If the reducibility is detected, there will be no optimal completion under the current partial solution. Therefore, the program can backtrack.

The reducibility can be checked by the following theorem:

Theorem 5.1.4 A partial solution* S is reducible if and only if there exist $j_1, j_2, \ldots, j_p$ for some $p > j$ such that

(1) $x_{j_i} \varepsilon S$ for $i = 1, 2, \ldots, p$,

(2) $x_{j_i} = 1$ for $i = 1, 2, \ldots, p$,

(3) $\vec{a}_{j_1} \leq \vec{a}_{j_2} + \vec{a}_{j_3} + \ldots + \vec{a}_{j_p}$,

where $\vec{a}_{j_i}$ is the column of A corresponding to variable x_{j_i} for $i = 1, 2, \ldots, p$.

Proof Suppose S is reducible and $x_{j_1}, x_{j_2}, \ldots, x_{j_p}$ are variables fixed to 1 in S. Then

(1) $x_{j_i} \varepsilon S$ for $i = 1, 2, \ldots, p$,

(2) $x_{j_i} = 1$ for $i = 1, 2, \ldots, p$.

Since S is reducible, there exists another partial solution S', which is obtained from S by setting a non-zero variable x_{j_1} in S to 0 and keeping all other variables unchanged, such that the constraints satisfied by S are also satisfied by S'. From the property that the constraints satisfied by S can be satisfied by S', it can be seen

* It does not matter whether S has fixed variables other than $x_{j_1}, x_{j_2}, \ldots, x_{j_p}$.

that at least one of a_{ij_2}, a_{ij_3}, ..., a_{ij_p} must be 1, if $a_{ij_1} = 1$. So p must be greater than 1 and $\vec{a}_{j_1} \leq \vec{a}_{j_2} + \ldots + \vec{a}_{j_p}$.

Conversely let us assume that there exist $j_1, j_2, \ldots, j_p$ for some $p > 1$ such that

(1) $x_{j_i} \in S$ for $i = 1, 2, \ldots, p$,

(2) $x_{j_i} = 1$ for $i = 1, 2, \ldots, p$,

(3) $\vec{a}_{j_1} \leq \vec{a}_{j_2} + \ldots + \vec{a}_{j_p}$.

Let $x_{j_1}, x_{j_2}, \ldots, x_{j_p}, x_{j_{p+1}}, \ldots, x_{j_{p+k}}$ be those variables fixed to 1 in S, and let S' be obtained from S by setting x_{j_1} to 0 and keeping all other variables in S unchanged. Now let us show that every constraint satisfied by S is also satisfied by S'.

Let $a_{i1} x_1 + a_{i2} + \ldots + a_{in} x_n \geq 1$ be a constraint satisfied by S. Then, by definition, $a_{ij_1} + a_{ij_2} + \ldots + a_{ij_{p+k}} \geq 1$. If $a_{ij_1} = 1$, then one of $a_{ij_2}, a_{ij_3}, \ldots, a_{ij_p}$ must be 1, since $\vec{a}_{ij_1} \leq \vec{a}_{ij_2} + \ldots + \vec{a}_{ij_p}$. If $a_{ij_1} = 0$, then $a_{ij_2} + a_{ij_3} + \ldots + a_{ij_{p+k}} \geq 1$. Therefore, the constraint $a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n \geq 1$ is satisfied by S'.

Q.E.D.

5.2 Excluding Relation Between Two Columns

If column $\vec{a}_j$ is not dominated by column $\vec{a}_i$, an E-set (E means excluding operation) of $\vec{a}_j$ with respect to $\vec{a}_i$ is defined to be

the set of rows which are covered by $\vec{a}_j$ but not by $\vec{a}_i$ and is denoted by E_{ij}.

<u>Example 5.2.1</u>

$$\text{If} \quad \vec{a}_i = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad \text{and} \quad \vec{a}_j = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \quad \text{then}$$

$E_{ij} = \{\vec{r}_3, \vec{r}_6\}$ and $E_{ji} = \{\vec{r}_2, \vec{r}_5\}$. ∎

The second new property of the minimal covering problem is stated in the following theorem and will be referred to as "the excluding property".

<u>Theorem 5.2.1</u> Let E_{ij} be the E-set of column $\vec{a}_j$ with respect to column $\vec{a}_i$ and let $\vec{x}$ be a feasible solution of (P) with $x_i = 0$, $x_j = 1$ and $x_{k_t} = 1$ for $t = 1, 2, \ldots, r$ (other variables are assigned either 0 or 1). If each row in E_{ij} is covered by some of the columns $\vec{a}_{k_1}, \vec{a}_{k_2}, \ldots, \vec{a}_{k_r}$, then $\vec{x}'$, which is obtained from $\vec{x}$ by replacing $x_i = 0$, $x_j = 1$ by $x_i = 1$, $x_j = 0$ and the remaining variables unchanged, is also a feasible solution of (P). The objective values of both solutions are the same.

<u>Proof</u> Let S be the partial solution with $x_i = 0$, $x_j = 1$ and $x_{k_t} = 1$ for t = 1, 2, ..., r, and let S' be the partial solution with $x_i = 1$, $x_j = 0$ and $x_{k_t} = 1$ for t=1, 2, ..., r. (S and S' have no other fixed variables.) Since each row in E_{ij} is covered by some of the columns $\vec{a}_{k_1}, \vec{a}_{k_2}, \ldots, \vec{a}_{k_r}$, each constraint of (P) satisfied by S is also satisfied by S'. Since $\vec{x}$ is a feasible completion of S (by the definitions of $\vec{x}$ and S), the constraints which are not satisfied by S, must be satisfied by the partial solution S", the set of the variables which are not fixed in S and are fixed to 0 or 1 by $\vec{x}$.

Since each constraint of (P) is satisfied either by S or S", and since each constraint satisfied by S is satisfied by S', each constraint of (P) is satisfied either by S' or by S". Since $\vec{x}$ is a solution obtained by assigning each variable according to S' and S" (by definition), $\vec{x}'$ is a feasible solution of (P).

From the definition of $\vec{x}'$, it is easy to see the objective value of $\vec{x}$ and $\vec{x}'$ are the same.

Q.E.D.

Suppose $\vec{a}_i$ and $\vec{a}_j$ are two columns covered by the row chosen at step M3.1 in the outline of the basic algorithm. (There may be more than 2 columns covered by this row, but consider only two of them as $\vec{a}_i$ and $\vec{a}_j$.) Let us consider the two subproblems corresponding to these two columns $\vec{a}_i$ and $\vec{a}_j$: one obtained by setting $x_i = 1$ and the other obtained by setting $x_i = 0$ and $x_j = 1$. After the subproblem with $x_i = 1$ has been implicitly enumerated but before the subproblem with $x_i = 0$ and $x_j = 1$ is considered, an E-set E_{ij} is

constructed. Then each time a free variable s_{k_r} is fixed to 1, E_{ij} is tested if each row in E_{ij} is covered by some column in $\{\vec{a}_{k_1}, \vec{a}_{k_2}, \ldots, \vec{a}_{k_r}\}$, where $x_{k_1}, x_{k_2}, \ldots, x_{k_r}$ are the variables fixed to 1 in the partial solution S (S may have other variables fixed to 1) and $\vec{a}_{k_t} \neq \vec{a}_j$ for $t = 1, 2, \ldots, r$. If each row in E_{ij} is covered, then for each feasible completion $\vec{x}$ of the current partial solution S, there is another feasible solution $\vec{x}'$, which is obtained from $\vec{x}$ by replacing $x_i = 0$, $x_j = 1$ by $x_i = 1$, $x_j = 0$ and the remaining variables unchanged, by Theorem 5.2.1. Since the objective values of $\vec{x}$ and $\vec{x}'$ are the same and $\vec{x}$ is a feasible solution examined before in the subproblem with $x_i = 1$, the objective value of $\vec{x}$ cannot be smaller than the value of the best solution obtained so far. So no feasible completion better than the best solution obtained so far can be found under the current partial solution. Thus, the program can backtrack*. Computation saved by this modification of the algorithm is illustrated in the dotted triangle in Figure 5.2.1.

5.3 Implementation

In order to implement the two properties stated in Sections 5.1 and 5.2, an array variable YY is introduced. For each partial solution S, YY is defined by

$$YY_i = \sum_{x_j \varepsilon S} a_{ij} x_j$$

---

* See Section 3.3.

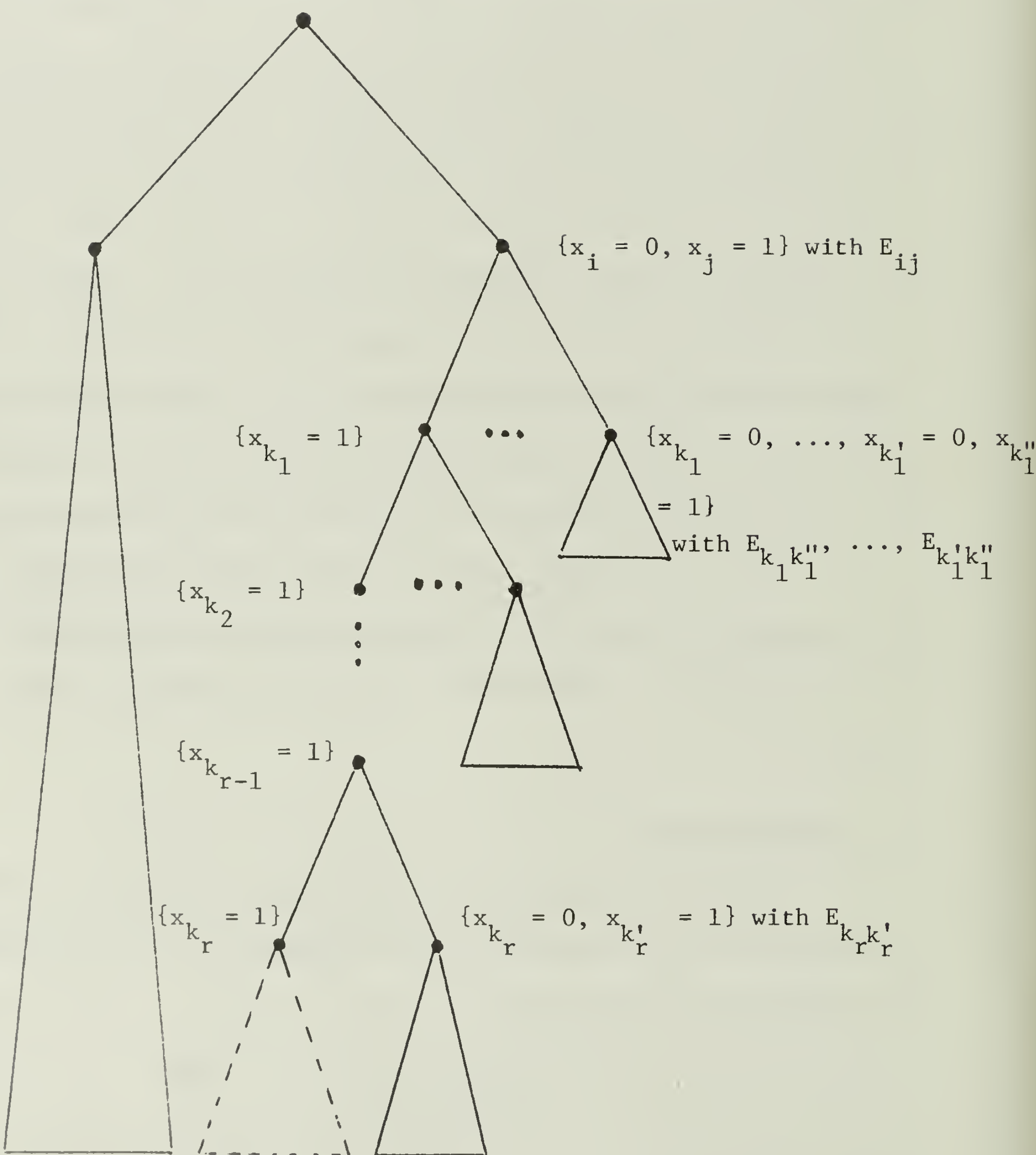


Figure 5.2.1 Computation saved by the checking of E_{ij}. -- The dotted triangle can be skipped when each row in E_{ij} is covered by some columns of $\vec{a}_{j_1}, \vec{a}_{j_2}, \ldots, \vec{a}_{j_r}$.

for $i = 1, 2, \ldots, m$, where a_{ij}'s are the elements of the given matrix. YY_i is called the current value of row i.

In the beginning, YY_i is initialized to 0 for each $i = 1, 2, \ldots, m$. It is updated by $YY_i = YY_i + a_{ij}$ for each i when a variable x_j is fixed to 1, and is updated by $YY_i = YY_i - a_{ij}$ for each i when a variable x_j with value 1 is set free.

Now the reducibility of a partial solution S can be checked as stated in the following theorem.

Theorem 5.3.1 A partial solution S is reducible if and only if there exists a $x_q \in S$ satisfying the following condition:

(1) $x_q = 1$, (5.3.1)

(2) $YY_k \geq 2$ for every k such that $a_{kq} = 1$, where YY_k is the current value of row k.

Proof From the definition of YY, $YY_k = \sum_{x_j \in S} a_{kj} x_j$ for $k = 1, 2, \ldots, m$. Let $x_{j_1} = x_q$ and all other non-zero variables in S be $x_{j_2}, \ldots, x_{j_p}$. Then $YY_k = a_{kj_1} + a_{kj_2} + \ldots + a_{kj_p}$ for $k = 1, 2, \ldots, m$. From (2) of the condition (5.3.1), $a_{kj_2} + \ldots + a_{kj_p} \geq 1$ whenever $a_{kj_1} = 1$, i.e., $\vec{a}_{j_1} \leq \vec{a}_{j_2} + \ldots + \vec{a}_{j_p}$ holds. Since at least one element of $\vec{a}_{j_1}$ is not 0, p must be greater than 1. By Theorem 5.1.4, S is reducible.

Conversely, let us assume S is reducible. By Theorem 5.1.4, there exist $x_{j_1}, x_{j_2}, \ldots, x_{j_p}$ for some $p \geq 1$ such that

(1) $x_{j_i} \in S$ for $i = 1, 2, \ldots, p$,

(2) $x_{j_i} = 1$ for $i = 1, 2, \ldots, p$,

(3) $\vec{a}_{j_1} \leq \vec{a}_{j_2} + \ldots + \vec{a}_{j_p}$.

From $\vec{a}_{j_1} \leq \vec{a}_{j_2} + \ldots \vec{a}_{j_p}$, it can be seen that one of

a_{kj_2}, a_{kj_3}, ..., a_{kj_p} must be 1, if $a_{kj_1} = 1$. In other words,

$$\sum_{i=1}^{p} a_{kj_i} \geq 2 \text{ for every } k \text{ such that } a_{kj_1} = 1. \qquad (5.3.2)$$

Let x_q be the x_{j_1}. Then $x_q \varepsilon S$ and $x_q = 1$. From (5.3.2), it can be seen that $YY_k = \sum_{i=1}^{p} a_{kj_i} \geq 2$ if $a_{kq} = 1$.

Q.E.D.

Whether each row in an E-set is covered by some of columns $\vec{a}_{k_1}$, $\vec{a}_{k_2}$, ..., $\vec{a}_{k_r}$ can be tested as stated in the following theorem.

Theorem 5.3.2 Let x_{k_1}, x_{k_2}, ..., x_{k_r} and x_j be variables* which are fixed to 1 in S. Each row in E_{ij}, the E-set of column $\vec{a}_j$ with respect to column $\vec{a}_{k_1}$, $\vec{a}_{k_2}$, ..., $\vec{a}_{k_r}$ if and only if

$YY_q \geq 2$ whenever row $\vec{r}_q$ is in E_{ij},

where YY_q is the current value of row $\vec{r}_q$.

Proof From the definition of YY,

$$YY_q = a_{qj} + \sum_{t=1}^{r} a_{qk_t} \quad \text{for } q = 1, 2, \ldots, m. \qquad (5.3.3)$$

From the definition of E_{ij}, it can be seen that $a_{qj} = 1$ if $\vec{r}_q$ is in E_{ij}. If each row in E_{ij} is covered by some column in $\{\vec{a}_{k_1}, \vec{a}_{k_2}, \ldots, \vec{a}_{k_r}\}$, then for each $\vec{r}_q$ in E_{ij}, at least one of a_{qk_1}, a_{qk_2}, ..., a_{qk_r} must

---

* It does not matter whether S has fixed variables set to 0, but S has no other fixed variables set to 1.

be one. So whenever $\vec{r}_q$ is in E_{ij}, $YY_q = a_{qj} + \sum_{i=1}^{r} a_{qk_i} \geq 2$.

Conversely, let us assume that $YY_q \geq 2$ whenever $\vec{r}_q$ is in E_{ij}. From equality (5.3.3),

$$a_{qj} + \sum_{t=1}^{r} a_{qk_t} \geq 2 \text{ whenever } \vec{r}_q \in E_{ij}. \qquad (5.3.4)$$

Since $\vec{r}_q \in E_{ij}$ implies $a_{qj} = 1$, it can be seen from (5.3.4) that

$$\sum_{t=1}^{r} a_{qk_t} \geq 1 \text{ whenever } \vec{r}_q \in E_{ij}. \qquad (5.3.5)$$

From (5.3.5), it is clear that at least one of a_{qk_1}, a_{qk_2}, ..., a_{qk_r} must be 1 whenever $\vec{r}_q \in E_{ij}$. Thus, $\vec{r}_q$ is covered by at one $\vec{a}_{k_1}$, $\vec{a}_{k_2}$, ..., $\vec{a}_{k_r}$ for each $\vec{r}_q$ in E_{ij}.

Q.E.D.

The following theorem shows a relation between the conditions stated in Theorems 5.3.1 and 5.3.2.

Theorem 5.3.3 Let S be a partial solution and YY_k be the current value of row k for each k. If there exists x_q in S which satisfies

(1) $x_q = 1$ (5.3.6)

(2) $YY_k \geq 2$ for each k such that $a_{kq} = 1$,

then each row of any E_{iq} such that $\vec{a}_q$ is not dominated by $\vec{a}_i$ satisfies

$$YY_k \geq 2 \text{ whenever } \vec{r}_k \text{ is in } E_{iq}. \qquad (5.3.7)$$

Proof Let $\vec{r}_k$ be a row in E_{iq}. Then $a_{kq} = 1$ by definition. Since $a_{kq} = 1$, $YY_k \geq 2$ is held by (2) of (5.3.6). Thus, (5.3.7) is proved.

Q.E.D.

Based on this theorem, the two properties introduced in this chapter can be implemented by testing only condition (5.3.7)

for an E-set E_{iq} whenever such an E-set is constructed.

For each subproblem, E-sets are constructed only when any of the subproblems, which are generated at the same time as the generation of this subproblem in step M3.2, have been implicitly enumerated. The number of E-sets constructed for this subproblem is the number of the subproblems which are generated at the same time as this subproblem and are already enumerated.

When a subproblem is generated by fixing a variable x_j to 1 for some j, sets of rows (i) or a set of rows (ii), in the following, are generated: (i) The E-sets $E_{i_1 j}$, $E_{i_2 j}$, ..., $E_{i_r j}$ for this subproblem, or (ii) the set of rows covered by the j-th column of the matrix A, if no E-set is constructed for this subproblem. These sets or this set is generated for testing the new backtrack conditions for this subproblem based on Corollary 5.1.3 and Theorem 5.2.1. All these sets for each level subproblem, which will be referred to as the <u>testing sets</u> in the following modified algorithm, are stored in a stack XQ along with the corresponding level numbers. When the program backtracks, those testing sets with level numbers greater than the current level numbers are deleted.

For each partial solution S, each testing set in the stack XQ is tested to see if this testing set satisfies $YY_k \geq 2$ for each k such that $\vec{r}_k$ is in it. If some testing set in XQ satisfies the above condition, then the program backtracks.

If only the "excluding property" is to be implemented, then only the E-sets for each subproblem are considered as the testing sets for that problem in the above discussion.

The whole algorithm with the two properties stated in Sections 5.1 and 5.2 incorporated is outlined as follows:

M1 Reduction.

Using the operation described in Section 2.1, reduce the constraint matrix as much as possible. Update the current partial solution and the current value YY_i for each row. If the matrix is reduced to a null matrix, go to Step M5.

M2 Bounding.

M2.1 Find a lower bound ZMIN of the subproblem under the current partial solution.

M2.2 Test if "ZBAR-ZMIN $\leq$ 0" is satisfied, where ZBAR is is the best value obtained so far. If it is satisfied go to M6.

M3 Branching.

M3.1 Choose a row $\vec{r}_i$ by some criterion.

M3.2 Based on the row chosen at M3.1, for each non-zero element a_{ij} in this row, generate a subproblem by fixing variable x_j to 1.

M3.3 Store indexes (j,-k) for all subproblems just generated in a stack XX, where j is the index of the branching variable and k is the level number of the subproblem. The problem corresponding to a column with the fewest non-zero elements is stored first.

M4 Next Subproblem.

M4.1 Get a variable from the top of stack XX and set it to 1. Update the current partial solution.

<u>M4.2</u> Update the current value YY_i for each row.

<u>M4.3</u> Test if any of the testing sets in XQ satisfies $YY_k \geq 2$ whenever $\vec{r}_k$ is in that set. If some of the testing sets satisfies the above condition, go to M6.

<u>M4.4</u> Generate testing sets for this subproblem, store them in a stack XQ, and go to M1.

<u>M5 Derivation Of A Feasible Solution</u>.

The current partial solution is a feasible solution. If this feasible solution is better than the best feasible solution obtained so far, keep this solution as the best feasible solution found so far.

<u>M6 Backtracking</u>.

<u>M6.1</u> Find the partial solution, in XSL, one level higher than the current subproblem and consider it as the current partial solution. In XSL, erase all partial solutions with level numbers greater than the level number of the current partial solution. If no partial solution is left in XSL, the given problem has been implicitly enumerated and the best solution obtained so far is an optimal solution.

<u>M6.2</u> Update the current value YY_i for each row i and the testing sets in XQ.

<u>M6.3</u> Retrieve the lower bound ZMIN, which was calculated previously at M2.1 or M6.7, for the current subproblem.

<u>M6.4</u> Test if "ZBAR-ZMIN $\leq$ 0" is satisfied. If it is satisfied go to M6.1.

<u>M6.5</u> Compare the level number LEVEL of the current partial solution with the level number LVT of the next subproblem to be considered in XX. If LVT <LEVEL + 1 go to M6.1. If LVT> LEVEL + 1, delete the next subproblem in XX and repeat M6.5.

<u>M6.6</u> Set the variable, based on which a subproblem has just been implicitly enumerated, to 0 and delete its corresponding column in the constraint matrix.

<u>M6.7</u> Calculate a lower bound ZMIN of the current subproblem and test if "ZBAR-ZMIN $\leq$ 0" is satisfied. If it is satisfied go to M6.1. Otherwise, go to M4.

<u>5.4 Some Computational Results</u>

The algorithm described at the end of Section 5.3 was coded in such a way that no reducibility of a partial solution is tested in the algorithm, i.e., only the E-sets for each subproblem are generated in step M4.4. This is because, from our experience with sample problems, the reducible condition stated in Theorem 5.3.1 was rarely satisfied by partial solutions in the algorithm outlined in the last section.

This algorithm (without checking the reducibility of a partial solution) was coded in FORTRAN. Some problems formulated from the logic minimization problem, obtained from literatures, or randomly generated by the author were tested by this program. Computational results are shown in Table 5.4.1. These results were obtained by

running programs on the IBM 360/75J computer using FORTRAN H compiler.

The number in the column under "d" shows the percentage of non-zero coefficients in the constraint matrix A of a problem, i.e., d = (No. of 1's in A) / (m × n).

The numbers in the columns under "m'" and "n'" are the numbers of rows and columns, respectively, left in the constraint matrix A after the program first went through the reduction procedure in

PROB. NO.	PROBLEM SIZE			PROBLEM SIZE AFTER THE FIRST REDUCTION		NO. OF ITER	NO. OF BKTRK	TIME IN SEC.
	m	n	d	m'	n'			
1	55	44	0.117	45	43	49	24	0.84
2	112	79	0.085	83	73	616	349	24.71
3	105	97	0.072	97	91	5079	3375	201.63
4	114	83	0.094	73	70	2254	1494	97.43
5	166	156	0.035	87	94	77	37	3.02
6	203	167	0.041	151	161	81877	57195	>6300*
7	35	15	0.20	35	15	159	77	1.24
8	117	27	0.11	117	27	6321	3063	94.14
9	60	60	0.062	43	50	47	24	0.93
10	60	80	0.065	52	75	350	191	7.7
11	30	90	0.07	30	80	62	26	1.14

Table 5.4.1 Some computational results by the algorithm in Section 5.3.

* It took 1046 seconds for the program to derive an optimal solution on the CDC Cyber 175 computer.

solving a problem. Numbers in the columns under "NO. OF ITER", "NO. OF BKTRK", and "TIME IN SEC." have the same meaning as those in their corresponding columns in Table 4.2.1, respectively.

Problems 1 through 6 are problems formulated from the logic minimization problem. Problem 1 is for minimizing the logic expression of a six-variable switching function; problems 2, 3, and 4 are for minimizing the logic expressions of seven-variable switching functions; problems 5 and 6 are for minimizing the logic expressions for eight-variable switching functions.

Problem 7 is the problem IBM 9 reported in [15], which has been used as a problem for comparison in many papers, such as [10, 11, 19, 20, 25]. Comparison of computational results for this problem is shown in Table 5.4.2.

PROGRAM	COMPUTER USED*	COMPUTATION TIME (in seconds)
Author's	IBM 360/75J	1.24
ILLIP [11]	IBM 360/75J	1.73
GEOFFRION'S [10]	IBM 7044	26.4
SHAPIRO'S [19]	IBM 360/65	20.2
ILP2 [20]	CDC 3600	75.1
ENUMER 8 [25]	CDC 6600	4.749
DSZ1IP [29]	CDC Cyber/175	1.236

Table 5.4.2 Comparison of computational results of the problem IBM 9

* The comparison of operational speeds of different computers is given later in this section

Problem 8 is the smaller of the two difficult problems reported in [24]. It is stated in [24] that these two problems may be used to measure the computational efficiencies of integer programming packages. Comparison of computational results of this problem is shown in Table 5.4.3.

PROGRAM	COMPUTER USED	COMPUTATION TIME (in seconds)
Author's	IBM 360/75I	94.14*
ENUMER 8 [25]	UNIVAC 1108	960

Table 5.4.3 Comparison of computational results of problem 8

Problems 9, 10, and 11 are problems randomly generated.

To make the comparison in Tables 5.4.2 and 5.4.3 more meaningful, operational speeds of different computers are shown in Table 5.4.4. All figures in this table except those in the last column are obtained from [26, 27]. The figures in the last column under "ESTIMATED RATIO" were given by the author according to the FIXED ADD/SUB time, the STORAGE CYCLE time and the number of register for each computer, which are listed in columns 2, 3 and 4, respectively. Only rough comparison can be made for the particular problems in Tables 5.4.2 and 5.4.3, since the estimated ratio may not be accurately applicable to these problems.

In order to see the computational improvement due to the checking of E-sets for subproblems, some problems were tested by both algorithms listed in Sections 3.3 and 5.3 (only "the excluding property" is implemented in the later algorithm). Comparison of computational results is shown in Table 5.4.5.

---

* The computation time is further reduced by considering the symmetric property of this problem, which is discussed in Chapter 7.

COMPUTER	FIXED ADD/SUB	STORAGE CYCLE	NO. OF REGISTER	ESTIMATED RATIO
IBM 360/75J	0.68	0.75	16	1
IBM 360/65J	1.4	0.75	16	≈ 1.5
IBM 360/50	4	2	16	≈ 4
IBM 7044	5	2	1	≈ 6.5
IBM 7094	2.8	1.4	1	≈ 4
CDC 3600	2.1	1.4	1	≈ 3.5
CDC 6600	0.3	1.0	8	≈ 0.9
UNIVAC 1108	0.75	0.75	16	≈ 1
CDC Cyber/175	—	—	—	≈ 0.16*

Table 5.4.4 Comparison of different computers

PROB. NO.†	PROBLEM SIZE		WITHOUT CHECKING E-SETS			WITH CHECKING E-SETS		
	M	N	NO. OF ITER	NO. OF BKTRK	TIME IN SEC.	NO. OF ITER	NO. OF BKTRK	TIME IN SEC.
1**	55	44	49	24	1.32	49	24	1.35
2**	112	79	841	477	54.18	616	394	39.92
3	105	97	7020	4107	275.42	5079	3375	201.63
4	114	83	3256	1879	126.49	2254	1194	97.49
5**	166	156	77	37	4.66	77	37	4.73
10**	60	80	378	198	12.22	350	191	11.60

Table 5.4.5 Comparison of two cases: with and without checking E-sets for the given problem

---

* No sufficient information about the operational speed of Cyber 175 is known. This ratio is estimated based on runing the same programs on two computers IBM 360/75I and CDC Cyber/175, by the author

** Results are obtained by using FORTRAN G compiler

† Problem numbers are those assigned in Table 5.4.1

From Table 5.4.5, it can be seen that the computational improvement due to the implementation of checking condition (5.3.7) for the E-sets in solving problems is roughly 30% for problems that need long computation time, such as problems 2, 3, and 4. It can also be seen from the table that checking condition (5.3.7) for the E-sets in solving problems does not improve the computational efficiency for problems which need only short computation time, such as problems 1 and 5. Since the amount of computation time spent in checking condition (5.3.7) for the E-sets is very small comparing to the time in solving problems (this can be seen from the computational results for problems 1 and 5 in Table 5.4.5), checking condition (5.3.7) for the E-sets is a very useful scheme for speeding up the enumeration.

6. AN HEURISTIC ALGORITHM FOR THE LARGE SCALE MINIMAL COVERING PROBLEM

The minimal covering problem (P) formulated for minimizing the logic expression of a complicated switching function with the number of switching variables greater than or equal to 8 usually has a large constraint matrix A. An example is the problem number 6 in Table 5.4.1, which is a problem formulated for minimizing the logic expresion of an eight-variable switching function and has a 206 by 167 constraint matrix. It is estimated in [8] that the number of prime implicants for a nine-variable switching function with 384 true vectors can be as large as 448. In other words, the size of the constraint matrix A of a minimal covering problem formulated for minimizing the logic expression of a nine-variable switching function can be as large as 384 by 448. To solve such a large minimal covering problem is beyond the capability of any existing computer program.

For handling large minimal covering problems, people developed heuristic algorithms. R. M. Bowan and E. S. McVey [22] published an algorithm for the fast approximate solution of large prime implicant tables. This algorithm has certain criterion to choose prime implicants and ends when a first feasible solution is found. R. Roth [23] published another heuristic method for the minimal covering problem. In this method, a feasible solution is further checked to see if any λ colums in the solution set* can be replaced by other $\lambda - 1$ columns not in this solution set.

---

* The solution set of a feasible solution is the set of columns whose corresponding variables are fixed to 1 in this feasible solution.

This chapter presents another heuristic algorithm for solving problems which require excessive computation time when they are to be solved by the implicit enumeration algorithm. This algorithm is a modification of the algorithm outlined in Section 5.3. It takes a reasonable amount of computation time and uses a reasonable amount of core memory in solving a large scale minimal covering problem. Computational results show high probability of obtaining optimal solutions for large scale problems by this algorithm.

6.1 The Heuristic Algorithm

The basic idea of the algorithm introduced in Section 5.3 is that when a problem is difficult to solve directly, it is decomposed into smaller subproblems and each subproblem is solved individually. A subproblem is further decomposed into smaller subproblems if it is still difficult to solve. When the number of subproblems decomposed from the given problem is large, then it requires a large amount of core memory to store all these decomposed subproblems and requires a large amount of computation time to solve all these subproblems. One way to reduce the amounts of the required core memory and the required computation time is to skip subproblems which have low probability of deriving any optimal solutions.

The decomposition of a problem into subproblems can be represented by decomposition tree, which shows which subproblem is decomposed from which subproblem. Each subproblem in a decomposition tree is associated with a level number LEVEL which indicates the level of this subproblem in this decomposition tree, counted from the root of the tree. Usually the scale of subproblems at upper levels is greater than the scale of those at lower levels. Since an optimal solution of

of a small scale problem usually can be obtained by a simple selection criterion, such as the one used in [22], an heuristic algorithm which decomposes large scale subproblems into small scale subproblems and finds a feasible solution for each small scale subproblem by a simple selection criterion is developed. This algorithm is a modification of the algorithm in Section 5.3, and is described by only listing the modifications of that algorithm.

MODIFICATION 1 - the modification of M4 step.

The M4 step of the algorithm in Section 5.3 is modified as in the following M4' step.

M4' Next subproblem.

M4'.1 Get a variable from the top of stack XX and set it to 1. Suppose the variable is x_j. Then delete all rows with their j-th element equal to 1. Update the current partial solution.

M4'.2 Update the current value YY_i for each row.

M4'.3 Test if any of the testing sets in XQ satisfies $YY_k \geq 2$ whenever $\vec{\gamma}_k$ is in that set. If some of the testing sets satisfies the above condition, go to M6.

M4'.4 Generate testing sets for this subproblem, and store them in a stack XQ.

M4'.5 If the level number of this subproblem is less than the level limit, a positive integer specified by a user, then go to M1.

M4'.6 Solve this subproblem by an heuristic procedure HEURISTIC, which will be described later, and then got to M5'. ■

The heuristic procedure HEURISTIC used in M4'.6 is described as follows:

PROCEDURE HEURISTIC:

H1 Using the three reduction operations in Section 3.2, reduce the matrix as much as possible. If the matrix is null, then the procedure is terminated, obtaining a feasible solution.

H2 For each remaining column i, calculate

$$w_i = \left(\frac{1}{v_{i_1}} + \frac{1}{v_{i_2}} + \ldots + \frac{1}{v_{i_r}} \right)^*,$$

where $i_1, i_2, \ldots, i_r$ are the indices of the remaining rows covered by column i and v_{i_k} is the number of non-zero elements on row i_k for $k = 1, 2, \ldots, r$.

H3 Choose the column i_o with the greatest w_{i_o} and fix x_{i_o} to 1. If there is a tie, the one with the smallest column index is chosen.

H4 Delete all rows covered by the column chosen at H3 and go to H1. ■

---

* This formula is similar to the one used in [22].

MODIFICATION 2 - the modification of M5 step.

The M5 step of the algorithm in Section 5.3 is modified as in the following M5' step.

M5' Derivation of a feasible solution.

M5'.1 A feasible solution is obtained either through M1 step or through M4'.6 step. If the value Z of this solution is greater than the objective value ZBAR of the best solution obtained so far, then go to M6.

M5'.2 Apply a transformation procedure TR, which will be outlined later in this section, to this feasible solution to derive a better feasible solution if possible.

M5'.3 Let Z' be the value of the feasible solution obtained by the procedure TR. If Z' is less than the value ZBAR, then the value of ZBAR is replaced by Z', and the best solution obtained so far is replaced by the solution obtained by the procedure TR.

M5'.4 Go to M6. ■

Before the transformation procedure TR is outlined, the concept of "covering weight" of a feasible solution is introduced. Let $\vec{x} = (x_1, x_2, \ldots, x_n)$ be a feasible solution of the problem (p). The covering weight of $\vec{x}$, denoted by $W(\vec{x})$, is defined as the number of i's such that $YY_i \geq 2$, where $YY_i = \sum_{j=1}^{n} a_{ij} x_j$.

Transformation procedure TR

Procedure TR for a feasible solution $\vec{x} = (x_1, x_2, \ldots, x_n)$ consists of the following steps:

T1 Calculate $YY_i = \sum_{i=1}^{n} a_{ij} x_j$ for each $i = 1, 2, \ldots, m$.

T2 Check if there exists any $x_j = 1$ such that $YY_i \geq 2$ whenever row i is covered by column j. If there exists such a x_j, then x_j is set to 0, and YY_i's are updated. The solution obtained by fixing x_j from 1 to 0 and the remaining all other variables unchanged is still a feasible solution.

T3 Try to transform the current feasible solution into another feasible solution with greater covering weight, by substituting $\{x_i = 0, x_j = 1\}$ for $\{x_i = 1, x_j = 0\}$ in the current feasible solution for each pair of variables x_i and x_j.

T4 If a new feasible solution with improved covering weight is obtained in step T3, then update YY_k for each $k = 1, 2, \ldots, m$, and go to step T2. Otherwise the procedure terminates. ■

In step T3, for each index i, there are only few candidates for j such that the new solution is still feasible after $\{x_i = 1, x_j = 0\}$ is replaced by $\{x_i = 0, x_j = 1\}$. A procedure for performing step T3 for each x_i based on this observation is described as follows.

T3.1 Find the first row $\vec{r}_k$ covered by column $\vec{a}_i$ such that $YY_k = 1$.

T3.2 The variables corresponding to columns covered by row $\vec{r}_k$ are candidates for the variable x_j in step T3. Check only those candidates to see if a new feasible solution with improved cover weight can be obtained. ■

From MODIFICATION 1, if the level limit specified is large enough not to be reached in solving a problem, then the best solution obtained is still guaranteed to be optimal.

6.2 Some Computational Results

Some problems formulated from the logic minimization problem and some medium scale problems constructed by the author were tested by a FORTRAN program of this heuristic algorithm. They were solved on CDC Cyber 175 computer. Computational results are shown in Table 6.2.1. The number in the column under "VAL" is the best value obtained under the level limit shown in column "LEVEL LIMIT". All other figures in this table are the same as those figures in Table 5.4.1, except that those results were obtained under different level limits shown in their corresponding columns. "-" in the table shows no test is made for that case. "∞" in the column under "LEVEL LIMIT" means no level limit is specified, and the best value obtained in this case is the minimal value of the problem.

From this table one can see that the optimal solutions of the 5 test problems can all be obtained in a reasonable amount of computation time by specifying the level limit to 6. From this observation, this heuristic algorithm could be very practical for solving large scale minimal covering problem if level limits are appropriately specified.

PROB. NO.*	2			5			6			10			11		
PROB. SIZE	m 112	n 79		m 166	n 156		m 203	n 167		m 60	n 80		m 30	n 90	
LEVEL LIMIT	NO. OF BKTRK	TIME IN SEC	VAL	NO. OF BKTRK	TIME IN SEC	VAL	NO. OF BKTRK	TIME IN SEC	VAL	NO. OF BKTRK	TIME IN SEC	VAL	NO. OF BKTRK	TIME IN SEC	VAL
1	1	0.11	19	1	0.11	45	1	0.27	40	1	0.06	16	1	0.04	13
2	4	0.17	19	---	---	---	4	0.27	40	2	0.06	16	5	0.09	13
3	16	0.88	19	4	0.11	45	14	0.28	40	6	0.13	16	17	0.2	13
4	47	1.30	19	---	---	---	45	5.69	40	---	---	---	---	---	---
5	114	3.0	18	---	---	---	129	12.0	39	---	---	---	---	---	---
6	---	---	---	---	---	---	360	29.89	38	---	---	---	---	---	---
∞	394	5.51	18	37	0.6	45	57195	1046	38	191	1.49	16	26	0.22	13

Table 6.2.1 Computational results of the heuristic algorithm in Section 6.1

*Problem numbers are those in Table 5.4.1.

7. SYMMETRIC MINIMAL COVERING PROBLEMS

The use of the symmetric property of the given switching function in solving the minimal covering problem formulated for minimizing the logic expression of that switching function is first noted in [6]. In this chapter, the symmetric property of the minimal covering problem is explored in detail, and the utilization of these properties in the enumeration algorithm for this problem is discussed. Procedures for utilizing these properties are developed based on the theory of finite permutation group. By applying these procedures in solving symmetric minimal covering problems, the computational improvement of more than 10 times was gained for some problems. Furthermore it was confirmed that utilization of the symmetric properties was crucial in solving computationally difficult problems such as those reported in papers [15, 24] and large-scale problems formulated from the logic minimization problem.

7.1 Symmetric Permutations

Let $X = \{x_1, x_2, \ldots, x_n\}$. A permutation $\eta = X \to X$ is said to be a symmetric permutation of the minimal covering problem (P) if it has the following property:

> if $(x_1, x_2, \ldots, x_n)$ is a feasible solution of the problem (P), then $(\eta(x_1), \eta(x_2), \ldots, \eta(x_n))$ is also a feasible solution of the problem (P).

Of course, both feasible solutions must yield the same value to (P).

Example 7.1.1. Let us consider the problem (P) with a constraint matrix

$$A = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} . \qquad (7.1.1)$$

Let η be a permutation defined on $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ as

$$\eta = \begin{cases} x_1 \longrightarrow x_3, \\ x_2 \longrightarrow x_4, \\ x_3 \longrightarrow x_5, \\ x_4 \longrightarrow x_6, \\ x_5 \longrightarrow x_1, \\ x_6 \longrightarrow x_2. \end{cases} \qquad (7.1.2.)$$

It is easy to see that $(x_1, x_2, x_3, x_4, x_5, x_6) = (1, 1, 1, 1, 0, 0)$ is a feasible solution of the problem (P) with A in (7.1.1). Permuting this feasible solution according to the permutation η, $(\eta(x_1), \eta(x_2), \eta(x_3), \eta(x_4), \eta(x_5), \eta(x_6)) = (x_3, x_4, x_5, x_6, x_1, x_2) = (1, 1, 0, 0, 1, 1)$. Then $(x_1, x_2, x_3, x_4, x_5, x_6) = (1, 1, 0, 0, 1, 1)$ is also a feasible solution of the problem (P). In order to prove that η is a symmetric permutation of this problem, we have to examine all feasible solutions of this problem and check if $(\eta(x_1), \eta(x_2), \ldots, \eta(x_6))$ (regarding it as $(x_1, x_2, \ldots, x_6)$) for each feasible solution $(x_1, x_2, \ldots, x_6)$. This is a cumbersome work. In Section 7.4, a

theorem (Theorem 7.4.1) which can be easily used to check whether or not a given permutation is symmetric will be given. ■

The minimal covering problem (P) is said to be symmetric if it has some symmetric permutations. If a given minimal covering problem is symmetric, then the symmetric property of this problem can be utilized in solving this problem by the implicit enumeration method as stated in the following theorem.

Theorem 7.1.1 Suppose η is a symmetric permutation of the minimal covering problem (P) and $x_i = \eta(x_j)$. Then in solving (P) by the implicit enumeration method, variable x_i can be fixed to 0 without losing a better feasible solution (a feasible solution better than the best one obtained so far) in the subproblem with x_j fixed to 0, if the subproblem with x_j fixed to 1 has already been implicitly enumerated.

Proof For any feasible solution $(x_1, x_2, \ldots, x_n)$ with $x_i = 1$, $(\eta(x_1), \eta(x_2), \ldots, \eta(x_n))$, where $\eta(x_j) = x_i = 1$, is also a feasible solution, since η is a symmetric permutation. Both feasible solutions have the same value w. Since the subproblem with x_j fixed to 1 has been already implicitly enumerated, the value w cannot be smaller than the value of the best solution obtained so far. So only the case with x_i fixed to 0 in the subproblem with x_j fixed to 0 has to be considered after the subproblem with x_j fixed to 1 has been considered.

Q.E.D.

Theorem 7.1.1 is illustrated by the figure shown in Figure 7.1.1. The dotted triangle in Figure 7.1.1 can be skipped in the enumeration by Theorem 7.1.1.

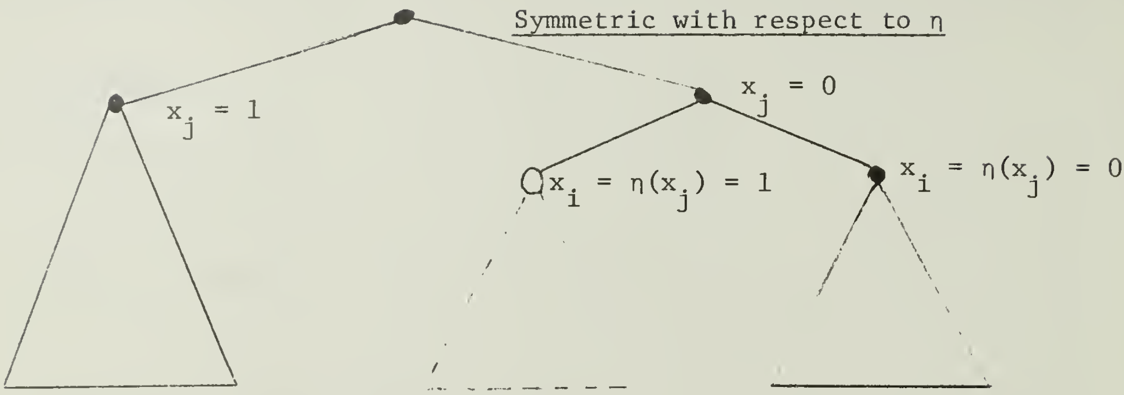


Figure 7.1.1. Illustration of Theorem 7.1.1 (The dotted triangle can be skipped).

For any two permutations η_1 and η_2 on $X = \{x_1, x_2, \ldots, x_n\}$, define a permutation $\eta_2 \circ \eta_1$ on X as

$$\eta_2 \circ \eta_1(x_i) = \eta_2(\eta_1(x_i))$$

for all x_i in X. A permutation $\overbrace{\eta \circ \eta \circ \ldots \circ \eta}^{i}$ on X is denoted by η^i.

Symmetric permutations have the following property.

Theorem 7.1.2 If η_1 and η_2 are two symmetric permutations of the minimal covering problem (P), then $\eta_2 \circ \eta_1$ is also a symmetric permutation of the problem (P).

Proof If $(x_1, x_2, \ldots, x_n)$ is a feasible equation of the problem (P), $(\eta_1(x_1), \eta_1(x_2), \ldots, \eta_1(x_n))$ is also a feasible solution of (P) since η_1 is symmetric. If $(\eta_1(x_1), \eta_1(x_2), \ldots, \eta_1(x_n))$ is a feasible solution of (P), $(\eta_2(\eta_1(x_1)), \eta_2(\eta_1(x_2)), \ldots, \eta_2(\eta_1(x_n))$ is also a feasible solution of (P), since η_2 is symmetric. Thus it can be concluded that if $(x_1, x_2, \ldots, x_n)$ is a feasible solution of (P), $(\eta_2(\eta_1(x_1)), \eta_2(\eta_1(x_2)), \ldots, \eta_2(\eta_1(x_n)))$ is also a feasible solution of (P), i.e., $\eta_2 \circ \eta_1$ is symmetric.

Q.E.D.

Corollary 7.1.3 If η is a symmetric permutation of the minimal covering problem (P), then η^i is also a symmetric permutation of the problem (P) for any given positive integer i.

Proof Since $\eta^k = \eta \circ \eta^{k-1}$ holds and η, η^{k-1} are symmetric permutations for each k, η^k is also a symmetric permutation of (P), by Theorem 7.1.2. Repeating this argument for increasingly greater i, the property holds for i.

Q.E.D.

From the above corollary, η^2, η^3, ... are all symmetric permutations of (P) if η is a symmetric permutation of (P). Since the number of different permutations of all variables in X is finite, there exists an integer α such that $\eta^\alpha = I$, where I is the identity permutation (i.e., the permutation which maps each variable to itself), as well known in group theory. Let α_o be the smallest positive integer such that $\eta^{\alpha_o} = I$. η is called a generator of the symmetric permutations η, η^2, ..., $\eta^{\alpha_o - 1}$ and η^i is said to be generated from η for each i = 1, 2, ..., α_o-1.

Example 7.1.2 Let us assume that the permutation η given in (7.1.2) is a symmetric permutation of the problem (P) with the constraint matrix (7.1.1). By Corollary 7.1.3,

$$\eta^2 : \begin{cases} x_1 \longrightarrow x_5, \\ x_2 \longrightarrow x_6, \\ x_3 \longrightarrow x_1, \\ x_4 \longrightarrow x_2, \\ x_5 \longrightarrow x_3, \\ x_6 \longrightarrow x_4, \end{cases}$$

is also a symmetric permutation. By definitions of η and η^2

$$\eta^3 = \eta^2 \circ \eta : \begin{cases} x_1 \longrightarrow x_1, \\ x_2 \longrightarrow x_2, \\ x_3 \longrightarrow x_3, \\ x_4 \longrightarrow x_4, \\ x_5 \longrightarrow x_5, \\ x_6 \longrightarrow x_6, \end{cases}$$

is the identity permutation. Thus η is a generator of symmetric permutations η and η^2. Since $\eta^2 \circ \eta^2 = \eta \circ \eta^3 = \eta$, and $\eta^2 \circ \eta^2 \circ \eta^2 = \eta^2 \circ \eta = \eta^3 = I$, η^2 is also a generator of symmetric permutations η^2 and η. ■

7.2 Symmetric Permutations of The Problem Formulated From The Logic Minimization Problem

Let $f(y_1, y_2, \ldots, y_t)$ be a switching function of variables* $y_1, y_2, \ldots y_t$. A permutation λ on $Y = \{y_1, y_2, \ldots, y_t\}$ is said to be a <u>symmetric permutation</u> of $f(y_1, y_2, \ldots, y_t)$ if

$$f(y_1, y_2, \ldots y_t) = f(\lambda(y_1), \lambda(y_2), \ldots, \lambda(y_t)).$$

A switching function f is said to be <u>λ-symmetric</u>** (or permutation symmetric) if there exists some symmetric permutations for this function. Notice that even if a switching function is λ-symmetric, the function is not necessarily symmetric** (partially or totally).

---

* Inequality variables are denoted with x_i's, whereas switching variables are denoted by y_i's.

** In switching theory, a function $f(y_1, y_2, \ldots, y_t)$ is symmetric in $y_1, y_2, \ldots, y_t$ if it is unchanged for every permutation of $y_1, y_2, \ldots, y_t$.

For a switching function $f(y_1, y_2, \ldots, y_t)$ and a permutation λ on $Y = \{y_1, y_2, \ldots, y_t\}$, $\lambda(f)(y_1, y_2, \ldots, y_t)$ is used to denote $f(\lambda(y_1), \lambda(y_2), \ldots, \lambda(y_t))$.

Example 7.2.1 Let us consider the switching function

$$f(y_1, y_2, y_3, y_4) = y_1 \cdot y_2 \cdot \bar{y}_3 \vee y_2 \cdot y_3 \cdot \bar{y}_4 \vee \bar{y}_1 \cdot y_3 \cdot y_4 \vee y_1 \cdot \bar{y}_2 \cdot y_4. \tag{7.2.1}$$

Let λ_1 be a permutation on $\{y_1, y_2, y_3, y_4\}$ defined as

$$\lambda_1 : \begin{cases} y_1 \longrightarrow y_2, \\ y_2 \longrightarrow y_3, \\ y_3 \longrightarrow y_4, \\ y_4 \longrightarrow y_1. \end{cases} \tag{7.2.2}$$

Then
$$\begin{aligned} \lambda_1(f)\ (y_1, y_2, y_3, y_4) &= f\ (\lambda_1(y_1), \lambda_1(y_2), \lambda_1(y_3), \lambda_1(y_4)) \\ &= f\ (y_2, y_3, y_4, y_1) \\ &= y_2 \cdot y_3 \cdot \bar{y}_4 \vee y_3 \cdot y_4 \cdot \bar{y}_1 \vee \bar{y}_2 \cdot y_4 \cdot y_1 \vee y_2 \cdot \bar{y}_3 \cdot y_1 \\ &= f(y_1, y_2, y_3, y_4). \end{aligned}$$

Thus λ_1 is a symmetric permutation of f. ■

Symmetric permutations of a switching function f have the following property.

Theorem 7.2.1 If λ_1 and λ_2 are two symmetric permutations of a switching function f, then the permutation $\lambda_1 \circ \lambda_2$ defined by $\lambda_1 \circ \lambda_2(y_i) = \lambda_1(\lambda_2(y_i))$ for $i = 1, 2, \ldots, t$ is also a symmetric permutation of f.

The proof of the above property is omitted here since it is just the same as that of Theorem 7.1.2. Similar to Corollary 7.1.3, it can be proved that if λ is a symmetric permutation of f, then $\lambda^i = \overbrace{\lambda \circ \lambda \circ \cdots \circ \lambda}^{i}$ is also a symmetric permutation of f for any positive

integer i. Since the number of different permutations of variables $y_1, y_2, \ldots, y_t$ is finite, there is a smallest positive integer α_o such that $\lambda^{\alpha_o} = I$, the identity permutation.

Let us consider Example 7.2.1 again. As can be easily seen,

$$\lambda_1^2 = \lambda_1 \circ \lambda_1 : \begin{cases} y_1 \longrightarrow y_3, \\ y_2 \longrightarrow y_4, \\ y_3 \longrightarrow y_1, \\ y_4 \longrightarrow y_2, \end{cases}$$

$$\lambda_1^3 = \lambda_1 \circ \lambda_1^2 : \begin{cases} y_1 \longrightarrow y_4, \\ y_2 \longrightarrow y_1, \\ y_3 \longrightarrow y_2, \\ y_4 \longrightarrow y_3, \end{cases}$$

are symmetric permutations of the function f defined in (7.2.1), and

$$\lambda^4 = \lambda \circ \lambda^3 : \begin{cases} y_1 \longrightarrow y_1, \\ y_2 \longrightarrow y_2, \\ y_3 \longrightarrow y_3, \\ y_4 \longrightarrow y_4, \end{cases}$$

is the identity permutation.

In the following, we shall show that corresponding to each symmetric permutation λ of a switching function f, there exists a symmetric permutation $\tilde{\lambda}$ of the minimal covering problem formulated for the logic minimization problem of f.

Define $\lambda(\bar{y}_i) = \overline{\lambda(y_i)}$ for $i = 1, 2, \ldots, t$.

Lemma 7.2.2 If λ is a symmetric permutation of $f(y_1, y_2, \ldots, y_t)$ and $q = Z_1 \cdot Z_2 \cdot \cdots \cdot Z_k$, where $Z_i = y_{j_i}$ or $\bar{y}_{j_i}$ for some $j_i \in \{1, 2, \ldots, t\}$,

is an implicant* of f; then $\lambda(q) = \lambda(Z_1)\cdot\lambda(Z_2)\cdot \cdots \cdot\lambda(Z_k)$ is also an implicant of f.

Proof Let us first prove this lemma for the case when f is a single-output switching function.

Since $q = Z_1\cdot Z_2\cdot \cdots \cdot Z_k$ is an implicant of f, f can be written as $f = f' \vee Z_1\cdot Z_2\cdot \cdots \cdot Z_k$ with some switching function $f'(y_1, y_2, \ldots, y_t)$. Since λ is a symmetric permutation of f, $f(y_1, y_2, \ldots, y_t) = \lambda(f)(y_1, y_2, \ldots, y_t) = f'(\lambda(y_1)\cdot\lambda(y_2), \ldots, \lambda(y_t)) \vee \lambda(Z_1)\cdot\lambda(Z_2)\cdot \cdots \cdot \lambda(Z_k)$. The above equalities show that $\lambda(Z_1)\cdot(Z_2)\cdot \cdots \cdot\lambda(Z_k)$ is also an implicant of $f(y_1, y_2, \ldots, y_t)$.

For the case when f is a multiple-output switching function, this lemma can be similarily proved.

Q.E.D.

Symmetric permutation λ of a switching function has the following property.

Theorem 7.2.3 If λ is a symmetric permutation of $f(y_1, y_2, \ldots, y_t)$ and $q = Z_1\cdot Z_2\cdot \cdots \cdot Z_k$, where $Z_i = y_{j_i}$ or $\bar{y}_{j_i}$ for some $j_i \varepsilon\{1, 2, \ldots, t\}$, is a prime implicant** of f, then $\lambda(q) = \lambda(Z_1)\cdot\lambda(Z_2)\cdot \cdots \cdot\lambda(Z_k)$ is also a prime implicant of f.

Proof Let us first prove this theorem for the case when f is a single-output switching function.

---

* either a single-output implicant or a multiple-output implicant.

** either a single-output prime implicant or a multiple-output prime implicant.

From Lemma 7.2.2, $\lambda(Z_1)\cdot\lambda(Z_2)\cdot \cdots \cdot\lambda(Z_k)$ is an implicant of f. If $\lambda(Z_1)\cdot\lambda(Z_2)\cdot \cdots \cdot\lambda(Z_k)$ is not a prime implicant of f, then there exists some term q' subsumed by $\lambda(Z_1)\cdot\lambda(Z_2)\cdot \cdots \cdot\lambda(Z_k)$ such that q' is an implicant of f. Since q' is subsumed by $\lambda(Z_1)\cdot\lambda(Z_2)\cdot \cdots \cdot\lambda(Z_k)$, $q' = \lambda(Z_{\ell_1})\cdot\lambda(Z_{\ell_2})\cdot \cdots \cdot\lambda(Z_{\ell_p})$ must hold with $\ell_1, \ell_2, \cdots, \ell_p \varepsilon\{1, 2, \cdots, k\}$. Let α be a positive integer such that λ^{α} is the identity permutation. Since $\lambda^{\alpha-1}$ is also a symmetric permutation (as it can be easily seen), and $q' = \lambda(Z_{\ell_1})\cdot\lambda(Z_{\ell_2})\cdot \cdots \cdot\lambda(Z_{\ell_p})$ is an implicant of f.

$$\begin{aligned}\lambda^{\alpha-1}(q') &= \lambda^{\alpha-1}(\lambda(Z_{\ell_1}))\cdot\lambda^{\alpha-1}(\lambda(Z_{\ell_2}))\cdot \cdots \cdot\lambda^{\alpha-1}(\lambda(Z_{\ell_p})),\\ &= \lambda^{\alpha}(Z_{\ell_1})\cdot\lambda^{\alpha}(Z_{\ell_2})\cdot \cdots \cdot\lambda^{\alpha}(Z_{\ell_p}),\\ &= Z_{\ell_1}\cdot Z_{\ell_2}\cdot \cdots \cdot Z_{\ell_p},\end{aligned}$$

is also an implicant of f by Lemma 7.2.2. Since $Z_1\cdot Z_2\cdot \cdots \cdot Z_k$ subsumes $Z_{\ell_1}\cdot Z_{\ell_2}\cdot \cdots \cdot Z_{\ell_p}$, $Z_1\cdot Z_2\cdot \cdots \cdot Z_k$ is not a prime implicant of f. This contradicts to that $Z_1\cdot Z_2\cdot \cdots \cdot Z_k$ is a prime implicant of f.

For the case when f is multiple-output, this theorem can be similarly proved.

Q.E.D.

Symmetric permutation λ has another property as stated in the following theorem.

Theorem 7.2.4 If q_i and q_j are two different prime implicants* of a switching function f and λ is a symmetric permutation of f, then $\lambda(q_i)$, and $\lambda(q_j)$ are two different prime implicants of f.

Proof Let us first prove the case when f is single-output.

---

* either single-output prime implicant or multiple-output prime implicant

From Theorem 7.2.3, $\lambda(q_i)$ and $\lambda(q_j)$ are prime implicants of f. If $\lambda(q_i) = \lambda(q_j)$, then both terms must have identical literals. Let literal $\lambda(Z_{i_k})$ of $\lambda(q_i)$ be equal to some literal $\lambda(Z_{j_1})$ of $\lambda(q_i)$. Since λ is a permutation, Z_{i_k} must be equal to Z_{j_1}. Thus each literal of q_i is equal to a literal of q_j, and each literal of q_j is equal to a literal of q_i. Consequently, $q_i = q_j$. This contradicts the assumption that q_i and q_j are two different prime implicants of f.

For the case when f is multiple-output, this theorem can be similarly proved, since a multiple-output prime implicant of a multiple-output function is defined as a single-output prime implicant of some product of those single-output functions of f.

Q.E.D.

From Theorems 7.2.3 and 7.2.4, each symmetric permutation λ of f defines a permutation on the set $Q = \{q_1, q_2, \cdots, q_n\}$ of all the prime implicants (or all the multiple-output prime implicants) of f as

$$\lambda(q_i) = \lambda(Z_1)\cdot\lambda(Z_2)\cdot \cdots \cdot\lambda(Z_k) \text{ if } q_i = Z_1\cdot Z_2\cdot \cdots \cdot Z_k.$$

Now a permutation $\tilde{\lambda}$ of the minimal covering problem formulated for minimizing the logic expression of a switching function f (See Section 2.1) is defined as

$$\tilde{\lambda}(x_j) = x_i \text{ if and only if } \lambda(q_i) = q_j. \qquad (7.2.3)$$

The symmetric property of $\tilde{\lambda}$ is shown in the following theorem.

<u>Theorem 7.2.5</u> Let $Q = \{q_1, q_2, \cdots, q_n\}$ be the set of all prime implicants of a single-output switching function f (or the set of all multiple-output prime implicants of a multiple-output switching function f), and λ be a symmetric permutation of f. Then the permutation $\tilde{\lambda}$,

defined by (7.2.3), of the minimal covering problem (P) formulated for minimizing the logic expression of f is a symmetric permutation.

Proof Let us first prove the case when f is single-output.

Let $(x_1, x_2, \cdots, x_n)$ be a feasible solution of (P), and $x_{i_1}, x_{i_2}, \cdots, x_{i_s}$ be all the variables which are fixed to 1 in this feasible solution. Then $f = q_{i_1} \vee q_{i_2} \vee \cdots \vee q_{i_s}$. Let $q_{j_1}, q_{j_2}, \cdots, q_{j_s}$ be the prime implicants of f such that

$$\lambda(q_{i_k}) = q_{j_k} \quad \text{for } k = 1, 2, \cdots, s. \tag{7.2.4}$$

Since λ is a symmetric permutation of f,

$$\begin{aligned} f(y_1, y_2, \cdots, y_t) &= f(\lambda(y_1), \lambda(y_2), \cdots, \lambda(y_t)), \\ &= \lambda(q_{i_1}) \vee \lambda(q_{i_2}) \vee \cdots \vee \lambda(q_{i_s}), \\ &= q_{j_1} \vee q_{j_2} \vee \cdots \vee q_{j_s}. \end{aligned}$$

The above equalities show that $\{q_{j_1}, q_{j_2}, \cdots, q_{j_s}\}$ is a feasible solution set of f. From (7.2.3) and (7.2.4), $(\tilde{\lambda}(x_1), \tilde{\lambda}(x_2), \cdots, \tilde{\lambda}(x_n))$ is a vector with $\tilde{\lambda}(x_{j_k}) = x_{j_k} = 1$ for $k = 1, 2, \cdots, s$. Since $\{q_{j_1}, q_{j_2}, \cdots, q_{j_s}\}$ is a feasible solution set and $\tilde{\lambda}(x_{j_k}) = 1$ for $k = 1, 2, \cdots, s$, $(\tilde{\lambda}(x_1), \tilde{\lambda}(x_2), \cdots \tilde{\lambda}(x_n))$ is also a feasible solution of (P).

For the case when f is multiple-output, this theorem can be similarly proved.

Q.E.D.

On the minimal covering problem formulated for minimizing the logic expression of a swtiching function f, the symmetric permutation obtained from a symmetric permutation λ of f is denoted by $\tilde{\lambda}$.

Example 7.2.2 The permutation λ_1 defined by (7.2.2) is a symmetric permutation of the switching function f defined in (7.2.1). All the prime implicants of f are: $q_1 = y_1 \cdot y_2 \cdot \bar{y}_3$, $q_2 = y_2 \cdot y_3 \cdot \bar{y}_4$, $q_3 = \bar{y}_1 \cdot y_3 \cdot y_4$, $q_4 = y_1 \cdot \bar{y}_2 \cdot y_4$, $q_5 = y_1 \cdot y_2 \cdot \bar{y}_4$, $q_6 = \bar{y}_1 \cdot y_2 \cdot y_3$, $q_7 = y_1 \cdot \bar{y}_3 \cdot y_4$, $q_8 = \bar{y}_2 \cdot y_3 \cdot y_4$. All the true vectors of this function are: $\vec{y}_1 = (1, 1, 0, 0)$, $\vec{y}_2 = (0, 1, 1, 0)$, $\vec{y}_3 = (1, 1, 1, 0)$, $\vec{y}_4 = (1, 0, 0, 1)$, $\vec{y}_5 = (1, 1, 0, 1)$, $\vec{y}_6 = (0, 0, 1, 1)$, $\vec{y}_7 = (1, 0, 1, 1)$, $\vec{y}_8 = (0, 1, 1, 1)$. The prime implicant table of this function is as follows:

$$A = \begin{array}{c} \\ \vec{y}_1 \\ \vec{y}_2 \\ \vec{y}_3 \\ \vec{y}_4 \\ \vec{y}_5 \\ \vec{y}_6 \\ \vec{y}_7 \\ \vec{y}_8 \end{array} \begin{array}{c} \begin{array}{cccccccc} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 \end{array} \\ \left(\begin{array}{c|c|c|c|c|c|c|c} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{array}\right) \end{array} . \qquad (7.2.5)$$

The minimal covering problem (P) formulated for the logic minimization problem of f is as follows:

$$\text{minimize } x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8$$

$$\text{subject to} \quad A \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad (7.2.6)$$

$$x_i = 0 \text{ or } 1, \text{ for } i = 1, 2, \cdots, 8.$$

The symmetric permutation $\tilde{\lambda}_1$ of this problem corresponding to the symmetric permutation λ_1 of f is

$$\tilde{\lambda}_1 : \begin{cases} x_1 \longrightarrow x_4, \\ x_2 \longrightarrow x_1, \\ x_3 \longrightarrow x_2, \\ x_4 \longrightarrow x_3, \\ x_5 \longrightarrow x_7, \\ x_6 \longrightarrow x_5, \\ x_7 \longrightarrow x_8, \\ x_8 \longrightarrow x_6. \end{cases} \tag{7.2.7}$$

Consider another permutation λ_2 defined on $\{y_1, y_2, y_3, y_4\}$

$$\lambda_2 : \begin{cases} y_1 \longrightarrow y_2, \\ y_2 \longrightarrow y_1, \\ y_3 \longrightarrow y_4, \\ y_4 \longrightarrow y_3. \end{cases} \tag{7.2.8}$$

Then $f(\lambda_2(y_1), \lambda_2(y_2), \lambda_2(y_3), \lambda_2(y_4)) = f(y_2\ y_1\ y_4\ y_3) = y_2 \cdot y_1 \cdot \bar{y}_4 \vee y_1 \cdot y_4 \cdot \bar{y}_3 \vee \bar{y}_2 \cdot y_4 \cdot y_1 \vee y_2 \cdot \bar{y}_1 \cdot y_3$. Finding all prime implicants of $f(y_2, y_1, y_4, y_3)$ by the iterated consensus method [30], all the prime implicants of $f(y_2, y_1, y_4, y_3)$ are exactly the same as those of $f(y_1, y_2, y_3, y_4)$. So $f(y_1, y_2, y_3, y_4) = f(\lambda_2(y_1), \lambda_2(y_2), \lambda_2(y_3), \lambda_2(y_4))$ holds and λ_2 is also a symmetric permutation of f. Corresponding to this symmetric permutation $\tilde{\lambda}_2$ of f, there is another symmetric permutation λ_2 defined as

$$\tilde{\lambda}_2 : \begin{cases} x_1 \longrightarrow x_5, \\ x_2 \longrightarrow x_7, \\ x_3 \longrightarrow x_8, \\ x_4 \longrightarrow x_6, \\ x_5 \longrightarrow x_1, \\ x_6 \longrightarrow x_4, \\ x_7 \longrightarrow x_2, \\ x_8 \longrightarrow x_3, \end{cases} \tag{7.2.9}$$

of the problem (7.2.6). ■

The following example shows that some minimal covering problems may have symmetric permutations that need more than one generator. The permutations $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ defined in (7.2.7) and (7.2.8) are symmetric permutations of the problem (7.2.6). From the definitions of $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$, it is easy to see that

(1) $\tilde{\lambda}_1^4 = \tilde{\lambda}_2^2 = I$, the identity permutation,

(2) $\tilde{\lambda}_1 \neq \tilde{\lambda}_2^i$, for any positive integer i,

(3) $\tilde{\lambda}_2 \neq \tilde{\lambda}_1^i$, for any positive integer i.

Thus the minimal covering problem of Example 7.2.2, has symmetric permutations that need more than one generator.

Now let us show a property of symmetric permutations of the problem obtained from the logic minimization problem of a switching function f. If λ_1 and λ_2 are two symmetric permutations of a switching function f, then $\lambda_1 \circ \lambda_2$ is also a symmetric permutation of f, by Theorem 7.2.1. Corresponding to $\lambda_1 \circ \lambda_2$ of f, there is the symmetric permutation $\widetilde{\lambda_1 \circ \lambda_2}$ of the problem (P) formulated for the logic minimization problem of f. $\widetilde{\lambda_1 \circ \lambda_2}$ has the following property.

Theorem 7.2.5 If λ_1 and λ_2 are two symmetric permutations of a switching function f, then $\widetilde{\lambda_1 \circ \lambda_2} = \tilde{\lambda}_2 \circ \tilde{\lambda}_1$, where $\tilde{\lambda}_1$, $\tilde{\lambda}_2$ and $\widetilde{\lambda_1 \circ \lambda_2}$ are symmetric permutations of the minimal covering problem formulated from the logic minimization problem of f corresponding to λ_1, λ_2 and $\lambda_1 \circ \lambda_2$, respectively.

Proof Let $Q = \{q_1, q_2, \cdots, q_n\}$ be the set of all prime implicants of f. From the definition of $\widetilde{\lambda_1 \circ \lambda_2}$,

$$\widetilde{\lambda_1 \circ \lambda_2}(x_j) = x_i \text{ if and only if } \lambda_1 \circ \lambda_2(q_i) = q_j. \qquad (7.2.10)$$

From the definitions of $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$,

$$\tilde{\lambda}_1(x_j) = x_\ell \text{ if and only if } \lambda_1(q_\ell) = q_j \qquad (7.2.11)$$

$$\text{and } \tilde{\lambda}_2(x_\ell) = x_i \text{ if and only if } \lambda_2(q_i) = q_\ell. \qquad (7.2.12)$$

By (7.2.10), (7.2.11), and (7.2.12), we have

$$\begin{aligned}
\tilde{\lambda}_2 \circ \tilde{\lambda}_1(x_j) = x_i &\text{ if and only if } \tilde{\lambda}_2(x_\ell) = x_i \text{ and } \tilde{\lambda}_1(x_j) = x_\ell \text{ for some } \ell, \\
&\text{if and only if } \lambda_2(q_i) = q_\ell \text{ and } \lambda_1(q_\ell) = q_j \text{ for some } \ell, \\
&\text{if and only if } \lambda_1 \circ \lambda_2(q_i) = q_j \\
&\text{if and only if } \widetilde{\lambda_1 \circ \lambda_2}(x_j) = x_i \qquad (7.2.13)
\end{aligned}$$

From (7.2.13), $\tilde{\lambda}_2 \circ \tilde{\lambda}_1 = \widetilde{\lambda_1 \circ \lambda_2}$

Q.E.D.

Example 7.2.3 The permutations λ_1 and λ_2 defined in (7.2.2) and (7.2.8) are symmetric permutations of the switching function f defined in (7.2.1). By definition of λ_1, λ_2 and $\lambda_2 \circ \lambda_1$,

$$\lambda_2 \circ \lambda_1 \left\{ \begin{array}{l} y_1 \longrightarrow y_1, \\ y_2 \longrightarrow y_4, \\ y_3 \longrightarrow y_3, \\ y_4 \longrightarrow y_2. \end{array} \right. \qquad (7.2.14)$$

The symmetric permutation $\widetilde{\lambda_2 \circ \lambda_1}$ on the problem (7.2.6) corresponding to $\lambda_2 \circ \lambda_1$ is

$$\widetilde{\lambda_2 \circ \lambda_1} : \begin{cases} x_1 \longrightarrow x_7, \\ x_2 \longrightarrow x_8, \\ x_3 \longrightarrow x_6, \\ x_4 \longrightarrow x_5, \\ x_5 \longrightarrow x_4, \\ x_6 \longrightarrow x_3, \\ x_7 \longrightarrow x_1, \\ x_8 \longrightarrow x_2, \end{cases} \tag{7.2.15}$$

by definition (7.2.3). From (7.2.7) and (7.2.9),

$$\tilde{\lambda}_1 \circ \tilde{\lambda}_2 : \begin{cases} x_1 \longrightarrow x_7, \\ x_2 \longrightarrow x_8, \\ x_3 \longrightarrow x_6, \\ x_4 \longrightarrow x_5, \\ x_5 \longrightarrow x_4, \\ x_6 \longrightarrow x_3, \\ x_7 \longrightarrow x_1, \\ x_8 \longrightarrow x_2, \end{cases}$$

which is exactly the same as (7.2.15). ■

Suppose the problem (P) is obtained from the logic minimization problem of a switching function f. The question arises whether, for each symmetric permutation of (P), there exists a corresponding symmetric permutation of f. The following example shows that the answer is negative.

<u>Example 7.2.4</u> Suppose that $f_1(y_1, y_2, y_3, y_4, y_5, y_6) =$ $y_1 \cdot y_2 \cdot \bar{y}_3 \cdot y_5 \cdot \bar{y}_6 \vee y_2 \cdot y_3 \cdot \bar{y}_4 \cdot y_5 \cdot \bar{y}_6 \vee \bar{y}_1 \cdot y_3 \cdot y_4 \cdot y_5 \cdot \bar{y}_6 \vee y_1 \cdot \bar{y}_2 \cdot y_4 \cdot y_5 \cdot \bar{y}_6$, $f_2(y_1, y_2, y_3, y_4, y_5, y_6) = y_1 \cdot \bar{y}_2 \cdot y_3 \cdot \bar{y}_5 \cdot y_6 \vee y_2 \cdot y_3 \cdot \bar{y}_4 \cdot \bar{y}_5 \cdot y_6 \vee \bar{y}_1 \cdot y_2 \cdot y_4 \cdot \bar{y}_5 \cdot y_6 \vee$ $y_1 \cdot \bar{y}_3 \cdot y_4 \cdot \bar{y}_5 \cdot y_6$, and $f(y_1, y_2, y_3, y_4, y_5, y_6) = f_1(y_1, y_2, y_3, y_4, y_5, y_6) \vee$ $f_2(y_1, y_2, y_3, y_4, y_5, y_6)$ are given. It is easy to see that

$$\lambda_1 : \begin{cases} y_1 \longrightarrow y_2, \\ y_2 \longrightarrow y_1, \\ y_3 \longrightarrow y_4, \\ y_4 \longrightarrow y_3, \\ y_5 \longrightarrow y_5, \\ y_6 \longrightarrow y_6, \end{cases} \tag{7.2.16}$$

is a symmetric permutation of f_1, and

$$\lambda_2 : \begin{cases} y_1 \longrightarrow y_3, \\ y_2 \longrightarrow y_4, \\ y_3 \longrightarrow y_1, \\ y_4 \longrightarrow y_2, \\ y_5 \longrightarrow y_5, \\ y_6 \longrightarrow y_6, \end{cases} \tag{7.2.17}$$

is a symmetric permutation of f_2. It is also easy to see that λ_1 is not a symmetric permutation of f_2 and λ_2 is not a symmetric permutation of f_1. Thus λ_1 and λ_2 are not symmetric permutations of f. Let us consider the logic minimization problem of the switching function f. All the prime implicants of f are: $q_1 = y_1 \cdot y_2 \cdot \bar{y}_3 \cdot y_5 \cdot \bar{y}_6$, $q_2 = y_2 \cdot y_3 \cdot \bar{y}_4 \cdot y_5 \cdot \bar{y}_6$, $q_3 = \bar{y}_1 \cdot y_3 \cdot y_4 \cdot y_5 \cdot \bar{y}_6$, $q_4 = y_1 \cdot \bar{y}_2 \cdot y_4 \cdot y_5 \cdot \bar{y}_6$, $q_5 = y_1 \cdot \bar{y}_3 \cdot y_2 \cdot y_5 \cdot \bar{y}_6$, $q_6 = y_2 \cdot y_3 \cdot \bar{y}_4 \cdot \bar{y}_5 \cdot y_6$, $q_7 = \bar{y}_1 \cdot y_2 \cdot y_4 \cdot \bar{y}_5 \cdot y_6$, $q_8 = y_1 \cdot \bar{y}_3 \cdot y_4 \cdot \bar{y}_3 \cdot y_6$, $q_9 = y_1 \cdot y_2 \cdot \bar{y}_4 \cdot y_5 \cdot y_6$, $q_{10} = \bar{y}_1 \cdot y_2 \cdot y_3 \cdot y_5 \cdot \bar{y}_6$, $q_{11} = y_1 \cdot \bar{y}_3 \cdot y_4 \cdot y_5 \cdot \bar{y}_6$, $q_{12} = \bar{y}_2 \cdot y_3 \cdot y_4 \cdot y_5 \cdot \bar{y}_6$, $q_{13} = y_1 \cdot y_3 \cdot \bar{y}_4 \cdot \bar{y}_5 \cdot y_6$, $q_{14} = \bar{y}_1 \cdot y_2 \cdot y_3 \cdot \bar{y}_5 \cdot y_6$,

$q_{15} = y_1 \cdot \bar{y}_2 \cdot y_4 \cdot \bar{y}_5 \cdot y_6$, $q_{16} = y_2 \cdot \bar{y}_3 \cdot y_4 \cdot y_5 \cdot y_6$. All the true vectors* of this function are: $\vec{y}_1 = (1, 1, 0, 0, 1, 0)$, $\vec{y}_2 = (0, 1, 1, 0, 1, 0)$, $\vec{y}_3 = (1, 1, 1, 0, 1, 0)$, $\vec{y}_4 = (1, 0, 0, 1, 1, 0)$, $\vec{y}_5 = (1, 1, 0, 1, 1, 0)$, $\vec{y}_6 = (0, 0, 1, 1, 1, 0)$, $\vec{y}_7 = (1, 0, 1, 1, 1, 0)$, $\vec{y}_8 = (0, 1, 1, 1, 1, 0)$, $\vec{y}_9 = (1, 0, 1, 0, 0, 1)$, $\vec{y}_{10} = (0, 1, 1, 0, 0, 1)$, $\vec{y}_{11} = (1, 1, 1, 0, 0, 1)$, $\vec{y}_{12} = (1, 0, 0, 1, 0, 1)$, $\vec{y}_{13} = (0, 1, 0, 1, 0, 1)$, $\vec{y}_{14} = (1, 1, 0, 1, 0, 1)$, $\vec{y}_{15} = (1, 0, 1, 1, 0, 1)$, $\vec{y}_{16} = (0, 1, 1, 1, 0, 1)$.

The prime implicant table for f is as follows:

$$A = \begin{array}{c|cccc:cccc:cccc:cccc|}
 & q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 & q_9 & q_{10} & q_{11} & q_{12} & q_{13} & q_{14} & q_{15} & q_{16} \\
\vec{y}_1 & 1 & 0 & 0 & 0 & & & & & 1 & 0 & 0 & 0 & & & & \\
\vec{y}_2 & 0 & 1 & 0 & 0 & & 0 & & & 0 & 1 & 0 & 0 & & 0 & & \\
\vec{y}_3 & 0 & 1 & 0 & 0 & & & & & 1 & 0 & 0 & 0 & & & & \\
\vec{y}_4 & 0 & 0 & 0 & 1 & & & & & 0 & 0 & 1 & 0 & & & & \\
\hdashline
\vec{y}_5 & 1 & 0 & 0 & 0 & & & & & 0 & 0 & 1 & 0 & & & & \\
\vec{y}_6 & 0 & 0 & 1 & 0 & & 0 & & & 0 & 0 & 0 & 1 & & 0 & & \\
\vec{y}_7 & 0 & 0 & 0 & 1 & & & & & 0 & 0 & 0 & 1 & & & & \\
\vec{y}_8 & 0 & 0 & 1 & 0 & & & & & 0 & 1 & 0 & 0 & & & & \\
\hdashline
\vec{y}_9 & & & & & 1 & 0 & 0 & 0 & & & & & 1 & 0 & 0 & 0 \\
\vec{y}_{10} & & 0 & & & 0 & 1 & 0 & 0 & & 0 & & & 0 & 1 & 0 & 0 \\
\vec{y}_{11} & & & & & 0 & 1 & 0 & 0 & & & & & 1 & 0 & 0 & 0 \\
\vec{y}_{12} & & & & & 0 & 0 & 0 & 1 & & & & & 0 & 0 & 1 & 0 \\
\hdashline
\vec{y}_{13} & & & & & 0 & 0 & 1 & 0 & & & & & 0 & 0 & 0 & 1 \\
\vec{y}_{14} & & 0 & & & 0 & 0 & 0 & 1 & & 0 & & & 0 & 0 & 0 & 1 \\
\vec{y}_{15} & & & & & 1 & 0 & 0 & 0 & & & & & 0 & 0 & 1 & 0 \\
\vec{y}_{16} & & & & & 0 & 0 & 1 & 0 & & & & & 0 & 1 & 0 & 0
\end{array} \quad . \qquad (7.2.18)$$

---

* It is coincident that the numbers of prime implicants and true vectors of f are the same in this example.

The logic minimization problem for f is as follows:

$$\text{minimize } x_1 + x_2 + \cdots + x_{16}$$

subject to

$$A \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{16} \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \qquad (7.2.19)$$

$$x_i = 0 \text{ or } 1 \text{ for } i = 1, 2, \cdots, 16.$$

It is easy to see that q_1, q_2, q_3, q_4, q_9, q_{10}, q_{11}, q_{12} are prime implicants of f_1 and q_5, q_6, q_7, q_8, q_{13}, q_{14}, q_{15}, q_{16} are prime implicants of f_2. Let η be a permutation of the problem (7.2.19) obtained by permuting $\{x_1, x_2, x_3, x_4, x_9, x_{10}, x_{11}, x_{12}\}$ according to $\tilde{\lambda}$, and permuting $\{x_5, x_6, x_7, x_8, x_{13}, x_{14}, x_{15}, x_{16}\}$ according to $\tilde{\lambda}_2$, i.e., η is defined as

$$\eta : \begin{cases} x_1 \longrightarrow x_{11}, \\ x_2 \longrightarrow x_{12}, \\ x_3 \longrightarrow x_{10}, \\ x_4 \longrightarrow x_9, \\ x_5 \longrightarrow x_5, \\ x_6 \longrightarrow x_6, \\ x_7 \longrightarrow x_7, \\ x_8 \longrightarrow x_8, \\ x_9 \longrightarrow x_4, \\ x_{10} \longrightarrow x_3, \\ x_{11} \longrightarrow x_1, \\ x_{12} \longrightarrow x_2, \\ x_{13} \longrightarrow x_{13}, \\ x_{14} \longrightarrow x_{14}, \\ x_{15} \longrightarrow x_{15}, \\ x_{16} \longrightarrow x_{16}. \end{cases} \qquad (7.2.20)$$

It will be proved in Section 7.4 that η is a symmetric permutation of the problem (7.2.19). But there is no symmetric permutation λ of the switching function f (i.e., a symmetric permutation among switching variables y_i's) corresponding to η of (7.2.20). ■

7.3 Complete Characterization Of Symmetric Permutations

The symmetric property of a minimal covering problem can be described by many symmetric permutations. If permutations $\eta_1, \eta_2, \cdots, \eta_h$ are used to describe the symmetric property, this symmetric minimal covering problem is said to be characterized by $\eta_1, \eta_2, \cdots, \eta_h$. A symmetric minimal covering problem is said to be more explicitly characterized by $\eta_1, \eta_2, \cdots, \eta_h$ than by symmetric permutations $\eta'_1, \eta'_2, \cdots, \eta'_h$ if each η'_i can be expressed as a concatenation of $\eta_1, \eta_2, \cdots, \eta_h$.

Example 7.3.1 Suppose η_1, η_2 and η_3 are defined by

$$\eta_1 : \begin{cases} x_1 \longrightarrow x_2, \\ x_2 \longrightarrow x_3, \\ x_3 \longrightarrow x_1, \\ x_4 \longrightarrow x_5, \\ x_5 \longrightarrow x_6, \\ x_6 \longrightarrow x_4, \end{cases} \qquad (7.3.1)$$

$$\eta_2 : \begin{cases} x_1 \longrightarrow x_2, \\ x_2 \longrightarrow x_3, \\ x_3 \longrightarrow x_1, \\ x_4 \longrightarrow x_4, \\ x_5 \longrightarrow x_5, \\ x_6 \longrightarrow x_6, \end{cases} \qquad (7.3.2)$$

$$\eta_3 : \begin{cases} x_1 \longrightarrow x_1, \\ x_2 \longrightarrow x_2, \\ x_3 \longrightarrow x_3, \\ x_4 \longrightarrow x_5, \\ x_5 \longrightarrow x_6, \\ x_6 \longrightarrow x_4, \end{cases} \tag{7.3.3}$$

are three symmetric permutations of the problem (P). Since $\eta_1 = \eta_2 \circ \eta_3$, the problem (P) is more explicitly characterized by η_2 and η_3 than by η_1. ∎

When a variable is fixed to 1 in a symmetric problem, some symmetric permutations may be destroyed and other symmetric permutations may be preserved. It will be shown later by Lemma 7.6.9 (Section 7.6) that if η is a symmetric permutation and $\eta(x_{k_o}) = x_{k_o}$, then the symmetric permutation η is preserved in the subproblem with x_{k_o} fixed to 1. As an example, let us consider Example 7.3.1 again. If x_1 is fixed to 1 in the problem (P), then symmetric permutations η_1 and η_2 are destroyed in the subproblem with x_1 fixed to 1 and symmetric permutation η_3 is preserved since $\eta_3(x_1) = x_1$. If the symmetric problem (P) is characterized by η_1 only (without η_2 and η_3) in Example 7.3.1, then the symmetric property in the subproblem with x_1 fixed to 1 cannot be detected.

A symmetric permutation η of the problem (P) is said to be <u>completely characterized</u> by symmetric permutations $\eta_1, \eta_2, \cdots, \eta_k$ of (P) if

(1) $\eta = \eta_{j_1} \circ \eta_{j_2} \circ \cdots \circ \eta_{j_r}$ for some $j_1, j_2, \cdots, j_r$ in $\{1, 2, \cdots, k\}$,

(2) η cannot be expressed as a concatenation of symmetric permutations other than η, η_1, η_2, $\cdots$, η_k and their concatenations.

Example 7.3.2 If η_1, η_2, η_3 in (7.3.1), (7.3.2), (7.3.3) and their concatenations in Example 7.3.1 are the only symmetric permutations of the problem (P), then

(1) $\eta_1 = \eta_2 \circ \eta_3$,

(2) η_1 cannot be expressed as a concatenation of symmetric permutations other than η_1, η_2, and η_3, since they are the only symmetric permutations of the problem (P), by assumption.

So η_1 is completely characterized by η_2 and η_3. ■

Now let us assume that a symmetric permutation η of the problem (P) is completley characterized by some other symmetric permutations η_1, η_2, $\cdots$, η_k. When a variable x_{k_o} is fixed to 1, some of η_1, η_2, $\cdots$, η_k are preserved and others are destroyed in the subproblem with x_{k_o} fixed to 1. If η can be expressed as $\eta_{j_1} \circ \eta_{j_2} \circ \cdots \circ \eta_{j_p}$ and η_{j_1}, η_{j_2}, $\cdots$, η_{j_p} are preserved in the subproblem with $x_{k_o} = 1$, then η is also preserved. If some of η_{j_1}, η_{j_2}, $\cdots$, η_{j_p} is destroyed in the subproblem with x_{k_o} fixed to 1, then it is not known whether η is still preserved in this subproblem since η may still be expressed as another concatenation of η_1, η_2, $\cdots$, η_k. If some of η_{j_1}, η_{j_2}, $\cdots$, η_{j_r} is destroyed in the subproblem for every concatenation $\eta_{j_1} \circ \eta_{j_2} \circ \cdots \circ \eta_{j_r}$ such that $\eta = \eta_{j_1} \circ \eta_{j_2} \circ \cdots \circ \eta_{j_r}$, then η is no longer preserved in the subproblem.

From the above discussion, it is completely dependent on the particular set of symmetric permutations $\eta_1, \eta_2, \cdots, \eta_k$ whether the permutation η is preserved in the subproblem with a variable x_{k_o} fixed to 1. Therefore, in solving the problem (P) by the implicit enumeration method, if the symmetric problem (P) is already characterized by $\eta_1, \eta_2, \cdots, \eta_k$, it is not necessary to add the symmetric permutation η to characterize the problem (P), since η is completely characterized by $\eta_1, \eta_2, \cdots, \eta_k$.

If symmetric permutations $\eta_1, \eta_2, \cdots, \eta_h$ of the problem (P) are completely characterized by symmetric permutations $\eta_1', \eta_2', \ldots, \eta_k'$, then the problem (P) is more explicitly characterized by $\eta_1', \eta_2', \cdots, \eta_k'$ than by $\eta_1, \eta_2, \cdots, \eta_h$. For any given symmetric permutations $\eta_1, \eta_2, \cdots, \eta_h$ of the problem (P), it is difficult to find symmetric permutations $\eta_1', \eta_2', \cdots, \eta_k'$ that completely characterize $\eta_1, \eta_2, \cdots, \eta_h$. But if it is possible to find symmetric permutations $\eta_1', \eta_2', \cdots, \eta_k'$ more explicitly characterizing this problem than $\eta_1, \eta_2, \cdots, \eta_h$, then by using $\eta_1', \eta_2', \cdots, \eta_k'$ as symmetric permutations characterizing this symmetric problem, the symmetric property of this problem will be more utilized in solving this problem by the implicit enumeration method.

The symmetric property of a λ-symmetric switching function can be described by many symmetric permutations. If $\lambda_1, \lambda_2, \cdots, \lambda_k$ are used to describe the symmetric property, then the λ-symmetric switching function is said to <u>be characterized</u> by $\lambda_1, \lambda_2, \cdots, \lambda_k$. A λ-symmetric switching function f is said to be <u>more explicitly characterized</u> by symmetric permutations $\lambda_1, \lambda_2, \cdots, \lambda_h$ (among switching variables) than by symmetric permutations $\lambda_1', \lambda_2', \cdots, \lambda_k'$ if each λ_i' can be expressed as a concatenation of $\lambda_1, \lambda_2, \cdots, \lambda_h$.

Example 7.3.3 The switching function f of (7.2.1) is more explicitly characterized by λ_1 of (7.2.2) and λ_2 of (7.2.8) than by $\lambda_2 \circ \lambda_1$ of (7.2.14). ■

From Theorem 7.2.5, it is easy to see that if a λ-symmetric switching function f is more explicitly characterized by symmetric permutations $\lambda_1, \lambda_2, \cdots, \lambda_h$ (among switching variables) than by $\lambda_1', \lambda_2', \cdots, \lambda_k'$, then the minimal covering problem (P) formulated from the logic minimization problem of f is more explicitly characterized by $\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_h$ than by $\tilde{\lambda}_1', \tilde{\lambda}_2', \cdots, \tilde{\lambda}_k'$, where $\tilde{\lambda}_1, \tilde{\lambda}_2, \cdots, \tilde{\lambda}_h$ and $\tilde{\lambda}_1, \tilde{\lambda}_2, \cdots, \tilde{\lambda}_k$ are symmetric permutations (among inequality variables) of the problem (P) corresponding to $\lambda_1, \lambda_2, \cdots, \lambda_h$ and $\lambda_1', \lambda_2', \cdots, \lambda_k'$, respectively.

Example 7.3.4 The minimal covering problem (7.2.6) is more explicitly characterized by symmetric permutations $\tilde{\lambda}_1$ of (7.2.7) and $\tilde{\lambda}_2$ of (7.2.9) than by symmetric permutation $\lambda_2 \circ \lambda_1$ of (7.2.15), since $\lambda_2 \circ \lambda_1 = \tilde{\lambda}_1 \circ \tilde{\lambda}_2$. ■

The complete characterization of a symmetric permutation λ of a λ-symmetric switching function f by another symmetric permutation $\lambda_1, \lambda_2, \cdots, \lambda_k$ of f is similarly defined as in the symmetric minimal covering problem case.

Example 7.3.5 The switching function f of (7.2.1) is λ-symmetric only in λ_1 of (7.2.2), λ_2 of (7.2.8) and their concatenations. The symmetric permutation $\lambda_2 \circ \lambda_1$ of (7.2.14) is completely characterized by λ_2 and λ_1 since $\lambda_2 \circ \lambda_1$ cannot be expressed as the concatenation of symmetric permutations other than λ_1, λ_2 and their concatenations. ■

The interchange of only two variables among $y_1, y_2, \cdots, y_t$ is called a transposition on $y_1, y_2, \cdots, y_t$. From group theory [34], any permutation on $y_1, y_2, \cdots, y_t$ (which is not necessarily a symmetric

permutation) can be expressed as a concatenation of transpositions on $y_1, y_2, \cdots, y_t$. From the definition of a symmetric switching function, each permutation on switching variables $y_1, y_2, \cdots, y_t$ of a symmetric switching function $f(y_1, y_2, \cdots, y_t)$ is a symmetric permutation of f. Since each transposition on $y_1, y_2, \cdots, y_t$ of a symmetric switch function $f(y_1, y_2, \cdots, y_t)$ is a permutation, each transposition on $y_1, y_2, \cdots, y_t$ of $f(y_1, y_2, \cdots, y_t)$ is also a symmetric permutation of f. Symmetric permutations of a symmetric switching function have the following property.

Theorem 7.3.1 If $f(y_1, y_2. \cdots, y_t)$ is a totally symmetric switching function then each symmetric permutation is completely characterized by transpositions on $y_1, y_2, \cdots, y_t$.

Proof By group theory, each permutation on $y_1, y_2. \cdots, y_t$ can be expressed as a concatenation of transpositions on $y_1, y_2, \cdots, y_t$. Let λ be a symmetric permutation on $y_1, y_2, \cdots, y_t$. Since each permutation on $y_1, y_2, \cdots, y_t$ can be expressed as a concatenation of transposition on $y_1, y_2, \cdots, y_t$,

(1) λ can be expressed as a concatenation of transpositions on $y_1, y_2, \cdots, y_t$.

(2) λ cannot be expressed as a concatenation of symmetric permutations other than transpositions on $y_1, y_2, \cdots, y_t$ and their concatenations, for the following reason:

If λ is expressed as a concatenation of symmetric permutations other than the transpositions of $y_1, y_2, \cdots, y_t$, then each symmetric permutation in this expression can further be expressed as a concatenation

of transpositions of $y_1, y_2, \cdots, y_t$. Thus λ is expressed in a concatenation of transpositions on $y_1, y_2, \cdots, y_t$ and their concatenations.

By definition, λ is completely characterized by transpositions on $y_1, y_2, \cdots, y_t$.

Q.E.D.

Let us show an example of how a permutation, which is not necessarily symmetric, can be expressed as a concatenation of transpositions.

Example 7.3.6

$$\lambda_1 : \begin{cases} y_1 \longrightarrow y_2, \\ y_2 \longrightarrow y_1, \\ y_3 \longrightarrow y_3, \end{cases} \quad \text{(i.e., exchange of } y_1 \text{ and } y_2\text{)}$$

$$\lambda_2 : \begin{cases} y_1 \longrightarrow y_3, \\ y_2 \longrightarrow y_2, \\ y_3 \longrightarrow y_1, \end{cases} \quad \text{(i.e., exchange of } y_1 \text{ and } y_3\text{)}$$

$$\lambda_3 : \begin{cases} y_1 \longrightarrow y_1, \\ y_2 \longrightarrow y_3, \\ y_3 \longrightarrow y_2, \end{cases} \quad \text{(i.e., exchange of } y_2 \text{ and } y_3\text{)}$$

are three transpositions on y_1, y_2, y_3. The permutation, which is not a transposition,

$$\lambda_4 : \begin{cases} y_1 \longrightarrow y_2, \\ y_2 \longrightarrow y_3, \\ y_3 \longrightarrow y_1, \end{cases}$$

can be expressed as $\lambda_4 = \lambda_2 \circ \lambda_1$ and the transposition λ_3 can also be expressed as $\lambda_3 = \lambda_1 \circ \lambda_2 \circ \lambda_1$. ■

Suppose λ is a symmetric permutation (among switching variables) of a switching function f and is completely characterized by symmetric permutations $\lambda_1, \lambda_2, \cdots, \lambda_k$ of f. Now the question arises whether the symmetric permutation $\tilde{\lambda}$, corresponding to λ, of the minimal covering problem (P) for the logic minimization problem of f is guaranteed to be completely characterized by $\tilde{\lambda}_1, \tilde{\lambda}_2, \cdots, \tilde{\lambda}_k$, where $\tilde{\lambda}_1, \tilde{\lambda}_2, \cdots, \tilde{\lambda}_k$ are the symmetric permutations of (P) corresponding to symmetric permutations $\lambda_1, \lambda_2, \cdots, \lambda_k$. Since, for some f, the problem (P) has symmetric permutations (among inequality variables) with no corresponding symmetric permutations (among switching variables) on f, the answer is negative as the following counter example shows.

<u>Example 7.3.7</u> The only symmetric permutation of the switching function f given in Example 7.2.4 are

$$\lambda_1 : \begin{cases} y_1 \longrightarrow y_2, \\ y_2 \longrightarrow y_1, \\ y_3 \longrightarrow y_4, \\ y_4 \longrightarrow y_3, \\ y_5 \longrightarrow y_5, \\ y_6 \longrightarrow y_6, \end{cases} \tag{7.3.4}$$

$$\lambda_2 : \begin{cases} y_1 \longrightarrow y_1, \\ y_2 \longrightarrow y_3, \\ y_3 \longrightarrow y_2, \\ y_4 \longrightarrow y_4, \\ y_5 \longrightarrow y_6, \\ y_6 \longrightarrow y_5, \end{cases} \tag{7.3.5}$$

and their concatenations. Since $\lambda_1 \neq \lambda_2^i$ for any positive integer i, λ_1 is completely specified by itself. The symmetric permutation corresponding

to λ_1 of the problem (7.2.19) is as follows:

$$\tilde{\lambda}_1 : \begin{cases} x_1 \longrightarrow x_9 , \\ x_2 \longrightarrow x_{11}, \\ x_3 \longrightarrow x_{12}, \\ x_4 \longrightarrow x_{10}, \\ x_5 \longrightarrow x_7 , \\ x_6 \longrightarrow x_8 , \\ x_7 \longrightarrow x_5 , \\ x_8 \longrightarrow x_6 , \\ x_9 \longrightarrow x_1 , \\ x_{10} \longrightarrow x_4 , \\ x_{11} \longrightarrow x_2 , \\ x_{12} \longrightarrow x_3 , \\ x_{13} \longrightarrow x_{16}, \\ x_{14} \longrightarrow x_{15}, \\ x_{15} \longrightarrow x_{14}, \\ x_{16} \longrightarrow x_{13}. \end{cases} \tag{7.3.6}$$

It will be proved later in Section 7.4 that

$$\eta_1 : \begin{cases} x_1 \longrightarrow x_9 , \\ x_2 \longrightarrow x_{11}, \\ x_3 \longrightarrow x_{12}, \\ x_4 \longrightarrow x_{10}, \\ x_5 \longrightarrow x_5 , \\ x_6 \longrightarrow x_6 , \\ x_7 \longrightarrow x_7 , \\ x_8 \longrightarrow x_8 , \\ x_9 \longrightarrow x_1 , \\ x_{10} \longrightarrow x_4 , \\ x_{11} \longrightarrow x_2 , \\ x_{12} \longrightarrow x_3 , \\ x_{13} \longrightarrow x_{13}, \\ x_{14} \longrightarrow x_{14}, \\ x_{15} \longrightarrow x_{15}, \\ x_{16} \longrightarrow x_{16}, \end{cases} \tag{7.3.7}$$

$$\eta_2 : \begin{cases} x_1 \longrightarrow x_1 , \\ x_2 \longrightarrow x_2 , \\ x_3 \longrightarrow x_3 , \\ x_4 \longrightarrow x_4 , \\ x_5 \longrightarrow x_{14}, \\ x_6 \longrightarrow x_{13}, \\ x_7 \longrightarrow x_{15}, \\ x_8 \longrightarrow x_{16}, \\ x_9 \longrightarrow x_9 , \\ x_{10} \longrightarrow x_{10}, \\ x_{11} \longrightarrow x_{11}, \\ x_{12} \longrightarrow x_{12}, \\ x_{13} \longrightarrow x_6 , \\ x_{14} \longrightarrow x_5 , \\ x_{15} \longrightarrow x_7 , \\ x_{16} \longrightarrow x_8 , \end{cases} \tag{7.3.8}$$

$$\eta_3 : \begin{cases} x_1 \longrightarrow x_1 , \\ x_2 \longrightarrow x_2 , \\ x_3 \longrightarrow x_3 , \\ x_4 \longrightarrow x_4 , \\ x_5 \longrightarrow x_{15}, \\ x_6 \longrightarrow x_{16}, \\ x_7 \longrightarrow x_{14}, \\ x_8 \longrightarrow x_{13}, \\ x_9 \longrightarrow x_9 , \\ x_{10} \longrightarrow x_{10}, \\ x_{11} \longrightarrow x_{11}, \\ x_{12} \longrightarrow x_{12}, \\ x_{13} \longrightarrow x_8 , \\ x_{14} \longrightarrow x_7 , \\ x_{15} \longrightarrow x_5 , \\ x_{16} \longrightarrow x_6 , \end{cases} \tag{7.3.9}$$

are symmetric permutations of the problem (7.2.19). From (7.3.6), (7.3.7), (7.3.8) and (7.3.9),

$$\tilde{\lambda}_1 = \eta_1 \circ \eta_2 \circ \eta_3. \qquad (7.3.10)$$

Thus $\tilde{\lambda}_1$ is derived from η_1, η_2 and η_3, and $\tilde{\lambda}_1$ is not completely specified by $\tilde{\lambda}_1$ itself. ■

7.4 A Necessary And Sufficient Condition For A Permutation To Be Symmetric

In this section, a necessary and sufficient condition for a given permutation to be symmetric is given.

For a given permutation η: $X \to X$, where $X = \{x_1, x_2, \cdots, x_n\}$, let $\eta(A)$ be the matrix obtained from A by permuting the columns of A according to the permutation η, i.e., the columns of $\eta(A)$ and A have the following relation:

$$\vec{b}_i = \vec{a}_j \text{ if and only if } \eta(x_i) = x_j, \qquad (7.4.1)$$

where $\vec{b}_i$ is the i-th column of $\eta(A)$ and $\vec{a}_j$ is the j-th column of A.

Example 7.4.1 Let the constraint matrix A of a given problem be

$$\begin{array}{c}
\underline{\text{column no.}} \quad \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\left(\begin{array}{c:c:c:c:c:c}
1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1
\end{array}\right)
\end{array} \qquad (7.4.2)$$

and a given permutation η on $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ be

$$\eta : \begin{cases} x_1 \longrightarrow x_2, \\ x_2 \longrightarrow x_3, \\ x_3 \longrightarrow x_1, \\ x_4 \longrightarrow x_4, \\ x_5 \longrightarrow x_6, \\ x_6 \longrightarrow x_5. \end{cases} \qquad (7.4.3)$$

Then

$$\begin{array}{rc} \underline{\text{column no.}} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \eta(A) = & \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \\ \underline{\text{old column no.}} & \begin{matrix} 2 & 3 & 1 & 4 & 6 & 5 \end{matrix} \end{array} \qquad (7.4.4)$$

■

A necessary and sufficient condition for a permutation η: $X \to X$ to be a symmetric permutation of the problem (P) is stated in the following theorem.

Theorem 7.4.1 A permutation η: $X \to X$ is a symmetric permutation of the problem (P) if and only if each row of A dominates some row of $\eta(A)$.

Proof First let us prove that if each row of A dominates some row of $\eta(A)$, then η is a symmetric permutation. Let $(x_1, x_2, \cdots, x_n)$ be a feasible solution of the problem (P). Then

$$A \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \qquad (7.4.5)$$

Rewrite inequality (7.4.5) as

$$\eta(A) \cdot \begin{bmatrix} \eta(x_1) \\ \eta(x_2) \\ \vdots \\ \eta(x_n) \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{7.4.6}$$

Since each row of A dominates some row of $\eta(A)$,

$$A \cdot \begin{bmatrix} \eta(x_1) \\ \eta(x_2) \\ \vdots \\ \eta(x_n) \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{7.4.7}$$

i.e., $(\eta(x_1), \eta(x_2), \cdots, \eta(x_n))$ is a feasible solution of (P). Thus η is symmetric.

Next let us prove that if there exists some row of A not dominating any row of $\eta(A)$, then η is not symmetric. Let the row of A not dominating any row of $\eta(A)$ be $(a_{i1}, a_{i2}, \cdots, a_{in})$, where $a_{ij_1}, a_{ij_2}, \cdots, a_{ij_r}$ be the non-zero elements (i.e., 1's). In the following let us construct a feasible solution $(x_1, x_2, \cdots, x_n)$ of the problem (P) such that $(\eta(x_1), \eta(x_2), \cdots, \eta(x_n))$ is not a feasible solution of (P).

Let us find a feasible solution of the following constraints:

$$\eta(A) \cdot \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{7.4.8}$$

$z_i = 0$ or 1 for $i = 1, 2, \ldots, n$.

Since $(a_{i1}, a_{i2}, \cdots, a_{in})$ does not dominate any row of $\eta(A)$, each row of $\eta(A)$ still has at least one non-zero element if columns $\vec{b}_{j_1}$, $\vec{b}_{j_2}$, $\cdots$, $\vec{b}_{j_r}$ (note that the ith elements in these columns are not necessarily 1's) have been deleted from $\eta(A)$. In other words, even if we set $z_{j_k} = 0$ for $k = 1, 2, \cdots, r$ in the constraints (7.4.8), constraints (7.4.8) is still feasible. Let $(z_1^*, z_2^*, \cdots, z_n^*)$ with $z_{j_k}^* = 0$ for $k = 1, 2, \ldots, r$ be a feasible solution of (7.4.8). So

$$\eta(A) \cdot \begin{pmatrix} z_1^* \\ z_2^* \\ \vdots \\ z_n^* \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \qquad (7.4.9)$$

Let η^{-1} be the inverse permutation of η and $(x_1^*, x_2^*, \cdots, x_n^*)$ be obtained from $(z_1^*, z_2^*, \cdots, z_n^*)$ by permuting $z_1^*, z_2^*, \cdots, z_n^*$ according to η^{-1}, i.e., $(x_1^*, x_2^*, \cdots, x_n^*)$ and $(z_1^*, z_2^*, \cdots, z_n^*)$ have the following relation

$$(\eta(x_1^*), \eta(x_2^*), \cdots, \eta(x_n^*) = (z_1^*, z_2^*, \cdots, z_n^*).$$

From (7.4.9),

$$A \cdot \begin{pmatrix} x_1^* \\ x_2^* \\ \vdots \\ x_n^* \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \qquad (7.4.10)$$

Inequality (7.4.10) shows that $(x_1^*, x_2^*, \cdots, x_n^*)$ is a feasible solution of the problem (P). Since the only non-zero elements in row i of the matrix A are a_{ij_1}, a_{ij_2}, $\cdots$, a_{ij_r} and $z_{j_1}^*$, $z_{j_2}^*$, $\cdots$, $z_{j_r}^*$ are 0,

$$A \cdot \begin{bmatrix} z_1^* \\ z_2^* \\ \vdots \\ z_n^* \end{bmatrix} \not\leq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} , \tag{7.4.11}$$

i.e., $(z_1^*, z_2^*, \cdots, z_n^*) = (\eta(x_1^*), \eta(x_2^*), \cdots, \eta(x_n^*))$ is not a feasible solution of (P).

Q.E.D.

Example 7.4.2 The constraint matrix of the problem in Example 7.1.1 is

$$A = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} . \tag{7.4.12}$$

The permutation η on this problem is

$$\eta : \begin{cases} x_1 \longrightarrow x_3, \\ x_2 \longrightarrow x_4, \\ x_3 \longrightarrow x_5, \\ x_4 \longrightarrow x_6, \\ x_5 \longrightarrow x_1, \\ x_6 \longrightarrow x_2. \end{cases} \tag{7.4.13}$$

Then, by the definition of $\eta(A)$,

$$\eta(A) = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} . \qquad (7.4.14)$$

Comparing matrices A and $\eta(A)$, it is easy to see that each row of A dominates some row of $\eta(A)$, so η is a symmetric permutation of this problem. ■

Now it can be proved that the permutations η, η_1, η_2, η_3 defined in (7.2.20), (7.3.7), (7.3.8), and (7.3.9) are symmetric permutations of the problem defined in (7.2.19). From definition,

$$\eta(A) = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \end{array} \left[\begin{array}{cccc|cccc|cccc|cccc} 0 & 0 & 0 & 1 & & & & & 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 1 & 0 & & 0 & & & 0 & 0 & 0 & 1 & & 0 & & \\ 0 & 0 & 0 & 1 & & & & & 0 & 0 & 0 & 1 & & & & \\ 1 & 0 & 0 & 0 & & & & & 1 & 0 & 0 & 0 & & & & \\ \hline 1 & 0 & 0 & 0 & & & & & 0 & 0 & 1 & 0 & & & & \\ 0 & 1 & 0 & 0 & & 0 & & & 0 & 1 & 0 & 0 & & 0 & & \\ 0 & 1 & 0 & 0 & & & & & 1 & 0 & 0 & 0 & & & & \\ 0 & 0 & 1 & 0 & & & & & 0 & 1 & 0 & 0 & & & & \\ \hline & & & & 1 & 0 & 0 & 0 & & & & & 1 & 0 & 0 & 0 \\ & 0 & & & 0 & 1 & 0 & 0 & & 0 & & & 0 & 1 & 0 & 0 \\ & & & & 0 & 1 & 0 & 0 & & & & & 1 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 1 & & & & & 0 & 0 & 1 & 0 \\ \hline & & & & 0 & 0 & 1 & 0 & & & & & 0 & 0 & 0 & 1 \\ & 0 & & & 0 & 0 & 0 & 1 & & 0 & & & 0 & 0 & 0 & 1 \\ & & & & 1 & 0 & 0 & 0 & & & & & 0 & 0 & 1 & 0 \\ & & & & 0 & 0 & 1 & 0 & & & & & 0 & 1 & 0 & 0 \end{array} \right] , \qquad (7.4.15)$$

$$
\eta_1(A) = \begin{array}{r} 1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14\\15\\16 \end{array}
\left[\begin{array}{cccc|cccc|cccc|cccc}
1&0&0&0& & & & &1&0&0&0& & & & \\
0&0&0&1& &0& & &0&0&1&0& &0& & \\
1&0&0&0& & & & &0&0&1&0& & & & \\
0&1&0&0& & & & &0&1&0&0& & & & \\
\hline
0&1&0&0& & & & &1&0&0&0& & & & \\
0&0&1&0& &0& & &0&0&0&1& &0& & \\
0&0&1&0& & & & &0&1&0&0& & & & \\
0&0&0&1& & & & &0&0&0&1& & & & \\
\hline
 & & & &1&0&0&0& & & & &1&0&0&0\\
 &0& & &0&1&0&0& &0& & &0&1&0&0\\
 & & & &0&1&0&0& & & & &1&0&0&0\\
 & & & &0&0&0&1& & & & &0&0&1&0\\
\hline
 & & & &0&0&1&0& & & & &0&0&0&1\\
 &0& & &0&0&0&1& &0& & &0&0&0&1\\
 & & & &1&0&0&0& & & & &0&0&1&0\\
 & & & &0&0&1&0& & & & &0&1&0&0
\end{array}\right], \tag{7.5.16}
$$

$$
\eta_2(A) = \begin{array}{r} 1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14\\15\\16 \end{array}
\left[\begin{array}{cccc|cccc|cccc|cccc}
1&0&0&0& & & & &1&0&0&0& & & & \\
0&1&0&0& &0& & &0&1&0&0& &0& & \\
0&1&0&0& & & & &1&0&0&0& & & & \\
0&0&0&1& & & & &0&0&1&0& & & & \\
\hline
1&0&0&0& & & & &0&0&1&0& & & & \\
0&0&1&0& &0& & &0&0&0&1& &0& & \\
0&0&0&1& & & & &0&0&0&1& & & & \\
0&0&1&0& & & & &0&1&0&0& & & & \\
\hline
 & & & &0&1&0&0& & & & &0&1&0&0\\
 &0& & &1&0&0&0& &0& & &1&0&0&0\\
 & & & &0&1&0&0& & & & &1&0&0&0\\
 & & & &0&0&1&0& & & & &0&0&0&1\\
\hline
 & & & &0&0&0&1& & & & &0&0&1&0\\
 &0& & &0&0&0&1& &0& & &0&0&0&1\\
 & & & &0&0&1&0& & & & &0&1&0&0\\
 & & & &1&0&0&0& & & & &0&0&1&0
\end{array}\right], \tag{7.4.17}
$$

$$\eta_3(A) = \begin{array}{r} 1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14\\15\\16 \end{array} \left[\begin{array}{cccccccccc}
1 & 0 & 0 & 0 & & 1 & 0 & 0 & 0 & \\
0 & 1 & 0 & 0 & & 0 & 1 & 0 & 0 & \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & & 0 & 0 & 1 & 0 & \\
\hline
1 & 0 & 0 & 0 & & 0 & 0 & 1 & 0 & \\
0 & 0 & 1 & 0 & & 0 & 0 & 0 & 1 & \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & & 0 & 1 & 0 & 0 & \\
\hline
 & 0 & 0 & 0 & 1 & & 0 & 0 & 1 & 0 \\
 & 0 & 0 & 1 & 0 & & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
 & 1 & 0 & 0 & 0 & & 1 & 0 & 0 & 0 \\
\hline
 & 0 & 1 & 0 & 0 & & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 0 & 0 & & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 & 0 & 0 & 1 & 0 & & 0 & 1 & 0 & 0
\end{array}\right] . \quad (7.4.18)$$

Comparing the matrices A of (7.2.18), $\eta(A)$ of (7.4.16), $\eta_1(A)$ of (7.4.17), $\eta_2(A)$ of (7.4.17) and $\eta_3(A)$ of (7.4.18), we can see that each row of A dominates one row in each of $\eta(A)$, $\eta_1(A)$, $\eta_2(A)$ and $\eta_3(A)$. Table 7.4.1 shows which row in $\eta(A)$, $\eta_1(A)$, $\eta_2(A)$, and $\eta_3(A)$ is dominated by row i of A for each i. For example, the sixth row of $\eta(A)$ is dominated by the second row of A.

A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\eta(A)$	4	6	7	1	5	2	3	8	9	10	11	12	13	14	15	16
$\eta_1(A)$	1	4	5	2	3	6	8	7	9	10	11	12	13	14	15	16
$\eta_2(A)$	1	2	3	4	5	6	7	8	10	9	11	13	12	14	16	15
$\eta_3(A)$	1	2	3	4	5	6	7	8	12	13	14	9	10	11	15	16

Table 7.4.1 Row domination relations among A and $\eta(A)$, $\eta_1(A)$, $\eta_2(A)$, η_3(A

Other properties of a symmetric permutation of a problem are given in the following theorems.

Theorem 7.4.2 If η is a symmetric permutation of the problem (P), then each row of $\eta(A)$ dominates some row of A.

Proof Assume $\eta \neq I$, where I is the identity permutation (If $\eta = I$, this theorem is trivial). Let q be an integer such that $\eta^q = I$. By Corollary 7.1.3, η^{q-1} is a symmetric permutation of (P). Now rewrite the problem (P) as follows:

$$\text{minimize } z_1 + z_2 + \cdots + z_n$$

subject to

$$A' \cdot \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

$$z_i = 0 \text{ or } 1 \text{ for } i = 1, 2, \cdots, n,$$

where $A' = \eta(A)$ and $z_i = \eta(x_i)$ for $i = 1, 2, \cdots, n$. Since η^{q-1} is a symmetric permutation of (P), each row of $A' = \eta(A)$ dominates some row of $\eta^{q-1}(A') = \eta^{q-1}(\eta(A)) = \eta^q(A) = A$ by Theorem 7.4.1.

Q.E.D

One can compare (7.4.12) and (7.4.14), and see that each row of $\eta(A)$ also dominates some row of A.

Theorem 7.4.3 If there is no row of A dominated by another row of A, then $\eta: X \to X$ is a symmetric permutation of the problem (P) if and only if for each row of A, there exists the identical row of $\eta(A)$.

Proof Only the "only if" case has to be proved, since the case of "if" is obvious from Theorem 7.4.1.

Assume $\vec{\gamma}_i$ is a row in A with no identical row in $\eta(A)$. Since η is symmetric, there exists a row $\vec{\gamma}'_j$ in $\eta(A)$ such that

$$\text{(1)}\quad \vec{\gamma}_i \text{ dominates } \vec{\gamma}'_j, \tag{7.4.19}$$

$$\text{(2)}\quad \vec{\gamma}_i \neq \vec{\gamma}_j \tag{7.4.20}$$

By Theorem 7.4.2, there exists some row $\vec{\gamma}_k$ in A dominated by $\vec{\gamma}'_j$. From (7.4.19) and (7.4.20), $i \neq k$ and $\vec{\gamma}_i$ dominates $\vec{\gamma}_k$ in A, which contradicts the assumption that there is no row of A dominated by another row of A.

Q.E.D.

From Theorem 7.4.3, it is easy to obtain the following corollary.

Corollary 7.4.4 If there is no row of A dominated by another row of A and if η is a symmetric permutation of the problem (P), then there is one-to-one correspondence of identity between the rows of A and those of $\eta(A)$.

Example 7.4.3 Let us reconsider the problem (P) with matrix (7.4.12). In this matrix, row 5 dominates row 3 and so there is no row in $\eta(A)$ identical to row 5 of A. If row 5 is deleted from matrix (7.4.12), then matrix (7.4.12) becomes

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \tag{7.4.21}$$

where no row is dominated by another. For the permutation η defined in (7.4.13),

$$\eta(A) = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \qquad (7.4.22)$$

Comparing the matrices A in (7.4.21) and $\eta(A)$ in (7.4.22), we can see that there is one-to-one correspondence between the rows of A and those of $\eta(A)$. ∎

7.5 Preservation Of A Symmetric Permutation During Program Backtracking

Let η be a permutation on $X = \{x_1, x_2, \ldots, x_n\}$, and r be the smallest positive integer such that $\eta^r(x_{k_o}) = x_{k_o}$. From Corollary 7.1.3 if η is a symmetric permutation of the problem (P), then $\eta, \eta^2, \cdots, \eta^{r-1}$ are also symmetric permutations on (P). After the subproblem with x_{k_o} fixed to 1 has been enumerated, each of $\eta(x_{k_o}), \eta^2(x_{k_o}) \ldots, \eta^{r-1}(x_{k_o})$ can be fixed to 0 in the subproblem with x_{k_o} fixed to 0 without losing a better feasible solution, by Theorem 7.1.1 (regard η^i and x_{k_o} as η and x_j in Theorem 7.1.1 respectively).

Theorem 7.5.1 Let η be a symmetric permutation of the problem (P). Then after the subproblem with x_{k_o} fixed to 1 has been enumerated, variables $\eta(x_{k_o}), \cdots, \eta^{r-1}(x_{k_o})$ can be fixed to 0 in the subproblem with $x_{k_o} = 0$ without losing a better feasible solution.

Proof Since

(1) the subproblem with variable x_{k_o}, $\eta(x_{k_o}), \cdots, \eta^{i-1}(x_{k_o})$ fixed to 0 is a subproblem of the subproblem

with x_{k_o} fixed to 0,

and (2) by Theorem 7.1.1, $\eta^i(x_{k_o})$ can be fixed to 0 in the subproblem with x_{k_o} fixed to 0 without losing a better feasible solution,

$\eta^i(x_{k_o})$ can be further fixed to 0 without losing a better feasible solution in the subproblem with x_{k_o}, $\eta(x_{k_o})$, $\cdots$, $\eta^{r-1}(x_{k_o})$ fixed to 0 for i = 1, 2, $\cdots$, r-1, if the subproblem with x_{k_o} fixed to 1 has been enumerated. Thus x_{k_o}, $\eta(x_{k_o})$, ..., $\eta^{r-1}(x_{k_o})$ can be fixed to 0 without losing a better feasible solution by repeatedly fixing $\eta^i(x_{k_o})$ to 0 in the subproblem with x_{k_o}, $\eta(x_{k_o})$, ..., $\eta^{i-1}(x_{k_o})$ fixed to 0 for i = 1, 2, $\cdots$, r-1, if the subproblem with x_{k_o} fixed to 1 has been enumerated.

Q.E.D.

Theorem 7.5.2 Let (P') be the subproblem obtained from (P) by fixing variables x_{k_o}, $\eta(x_{k_o})$, $\cdots$, $\eta^{r-1}(x_{k_o})$ to 0 and let $X' = X - \{x_{k_o}, \eta(x_{k_o}), \cdots, \eta^{r-1}(x_{k_o})\}$. If x_ℓ is a variable in X', then $\eta(x_\ell)$ is also a variable in X', where η is a permutation (not necessary be symmetric) on X = $\{x_1, x_2, \cdots, x_n\}$.

Proof If $\eta(x_\ell)$ is not a variable of X', then $\eta(x_\ell) = \eta^i(x_{k_o})$ for some $i > 0$. Then $x_\ell = \eta^{i-1}(x_{k_o})$, which shows that x_ℓ is not a variable of X'.

Q.E.D.

From Theorem 7.5.2, a permutation η' on X' can be defined by denoting $\eta(x_1)$ as $\eta'(x_1)$, i.e., $\eta'(x_1) = \eta(x_1)$, for all x_1 in x'. η' is said to be obtained from η by restricting it to (P') (i.e., restricting η from (P) to (P')). The following theorem shows that η' is a symmetric permutation of (P') if η is a symmetric permutation of (P).

Theorem 7.5.3 Let η be a symmetric permutation of problem (P) and r be the smallest positive integer such that $\eta^r(x_{k_o}) = x_{k_o}$. If (P') is the problem obtained from (P) by fixing variables x_{k_o}, $\eta(x_{k_o})$, $\cdots$, $\eta^{r-1}(x_{k_o})$ to 0 and η' is the permutation obtained from η by restricting it to (P'), then η' is a symmetric permutation of (P').

Proof Let A' be the constraint matrix of (P'). Since (P') is obtained from (P) by fixing x_{k_o}, $\eta(x_{k_o})$, ..., $\eta^{r-1}(x_{k_o})$ to 0, A' is a matrix obtained from A by deleting columns $\vec{a}_{k_o}$, $\vec{a}_{k_1}$, ..., $\vec{a}_{k_{r-1}}$ of A, where $x_{k_i} = \eta^i(x_{k_o})$ for i = 1, 2, ..., r-1. From the definition of $\eta(A)$, columns $\vec{b}_{k_o}$, $\vec{b}_{k_1}$, ..., $\vec{b}_{k_{r-1}}$ of $\eta(A)$ are columns $\vec{a}_{k_1}$, $\vec{a}_{k_2}$, ..., $\vec{a}_{k_{r-1}}$ $\vec{a}_{k_o}$ of A. From the definitions of η' and A', $\eta'(A')$ is obtained from $\eta(A)$ by deleting columns $\vec{b}_{k_o}$, $\vec{b}_{k_1}$, ..., $\vec{b}_{k_{r-1}}$.

Now let us show that each row of A' dominates some row of $\eta'(A')$ and thus η' is a symmetric permutation of (P'), by Theorem 7.4.1.

For each row $\vec{\gamma}_i'$ of A', its corresponding row $\vec{\gamma}_i$ in A (the row with the same index) dominates some row $\vec{\nu}_j$ in $\eta(A)$, since η is a symmetric permutation. Let $\vec{\nu}_j'$ be the row obtained by deleting k_o-th, k_1-th, ..., k_{r-1}-th elements from $\vec{\nu}_j$. Then $\vec{\nu}_j'$ is a row in $\eta'(A')$. Since $\vec{\gamma}_i'$ is a row obtained by deleting k_o-th, k_1-th, ..., k_{r-1}-th elements from $\vec{\gamma}_i$, $\vec{\gamma}_i'$ dominates $\vec{\nu}_j'$.

Q.E.D.

The discussion in this section is illustrated by Figure 7.5.1.

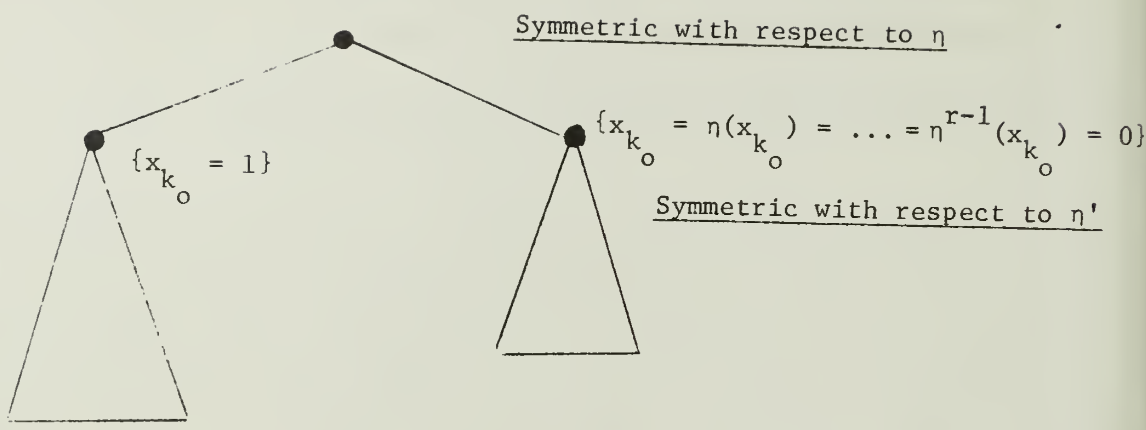


Figure 7.5.1 Illustration of a symmetric problem after the subproblem with x_{k_o} fixed to 1 is enumerated.

7.6 Preservation Of A Symmetric Permutation During The Three Reduction Operations

This section shows that a symmetric permutation η can be preserved during the following three reduction operations mentioned in Section 3.2:

(1) Deleting dominating rows in the constraint matrix.

(2) Deleting dominated columns in the constraint matrix and fixing their corresponding variables to 0.

(3) Fixing the variables corresponding to essential columns to 1 and deleting all rows covered by these columns.

Lemma 7.6.1 Let η be a symmetric permutation of the problem (P) and let (P1) be the problem obtained by deleting dominating rows from the constraint matrix of the problem (P). Then η is still a symmetric permutation of (P1).

Proof We only have to show that if $(x_1, x_2, \ldots, x_n)$ is a feasible solution

of (P1), then $(\eta(x_1), \eta(x_2), \ldots, \eta(x_n))$ is also a feasible solution of (P1).

From the definition of (P1), $(x_1, x_2, \ldots, x_n)$ is a feasible solution of (P1) if and only if $(x_1, x_2, \ldots, x_n)$ is a feasible solution of (P). Since η is a symmetric permution of (P), $(\eta(x_1), \eta(x_2), \ldots, \eta(x_n))$ is a feasible solution of (P). Thus $(\eta(x_1), \eta(x_2), \ldots, \eta(x_n))$ is a feasible solution of (P1).

Q.E.D.

From the above lemma, any given problem with some symmetric permutations can always be reduced to a problem with no dominating row in its constraint matrix and with the same symmetric permutations.

<u>Lemma 7.6.2</u> Suppose the constraint matrix A does not contain any dominating row. If $\eta(x_{k_o}) = x_{k_1}$ for a symmetric permutation η, then the numbers of non-zero elements in columns $\vec{a}_{k_o}$ and $\vec{a}_{k_1}$ of A are the same. Furthermore, if $\vec{a}_{k_1}$ dominates $\vec{a}_{k_o}$, then $\vec{a}_{k_1} = \vec{a}_{k_o}$.

<u>Proof</u> By definition, the k_o-th column of $\eta(A)$ is the k_1-th column of A. Since η is symmetric, the number of non-zero elements in the k_1-th column of A must be the same as that in the k_o-th column of A. Otherwise, there will not be a one-to-one correspondence of identity between the rows of A and (A), contradicting to the fact that η is symmetric. (This fact is due to Corollary 7.4.4). Thus the numbers of non-zero elements in $\vec{a}_{k_o}$ and $\vec{a}_{k_1}$ are the same.

If $\vec{a}_{k_1}$ also dominates $\vec{a}_{k_o}$, then $\vec{a}_{k_1} = \vec{a}_{k_o}$ because the numbers of of non-zero elements in these two columns are the same.

Q.E.D.

An example is the constraint matrix in (7.4.21), where the number of non-zero elements in column 1 is the same as that in column 3.

Lemma 7.6.3 Suppose the constraint matrix A contains no dominating row. If

(1) column $\vec{a}_{k_o}$ is dominated by some other column $\vec{a}_{s_o}$,

(2) $\eta(x_{k_o}) = x_{k_1}$ and $\eta(x_{s_o}) = x_{s_1}$ for some symmetric permutation η,

then column $\vec{a}_{k_1}$ is dominated by $\vec{a}_{s_1}$.

Proof By definition, the k_o-th column $\vec{b}_{k_o}$ of $\eta(A)$ is the k_1-th column $\vec{a}_{k_o}$ of A and the s_o-th column $\vec{b}_{s_o}$ of $\eta(A)$ is the k_1-th column $\vec{a}_{s_1}$ of A. If $\vec{a}_{k_1}$ is not dominated by $\vec{a}_{s_i}$ in A, then $\vec{b}_{k_o}$ is not dominated by $\vec{b}_{s_o}$ in $\eta(A)$. So there must exist some row $\vec{\nu}_i = (b_{i1}, b_{i2}, \ldots, b_{in})$ in $\eta(A)$ with $b_{ik_o} = 1$ and $b_{is_o} = 0$. Since $\vec{a}_{k_o}$ is dominated by $\vec{a}_{s_o}$ in A, there is no row in A identical to row $\vec{\nu}_i$ in $\eta(A)$. By Corollary 7.4.4, this contradicts the fact that η is symmetric.

Q.E.D.

Lemma 7.6.4 Suppose the constraint matrix A contains no dominating row and r is the smallest positive integer such that $\eta^r(x_k) = x_{k_o}$ for a symmetric permutation η. If (1) column $\vec{a}_{k_o}$ of A is dominated by some other column, and (2) $x_{k_1} = \eta(x_{k_o})$, $x_{k_2} = \eta^2(x_{k_o})$, ..., $x_{k_{r-1}} = \eta^{r-1}(x_{k_o})$, then $\vec{a}_{k_i}$ is also dominated by some other column for each i = 1, 2, ..., r.

Furthermore, if $\vec{a}_{k_o}$ is dominated by some column other than $\vec{a}_{k_o}, \vec{a}_{k_1}, \ldots, \vec{a}_{k_{r-1}}$, then $\vec{a}_{k_i}$ is dominated by some column other than $\vec{a}_{k_o}$,

$\vec{a}_{k_1}, \ldots, \vec{a}_{k_{r-1}}$.

Proof By Corollary 7.1.3., $\eta, \eta^2, \ldots, \eta^{r-1}$ are symmetric permutations. Since $\vec{a}_{k_o}$ is dominated by some other column, $\vec{a}_{k_i}$ is dominated by some other column for $i = 1, 2, \ldots, r-1$, by repeatedly applying Lemma 7.6.3.

Furthermore, if $\vec{a}_{k_o}$ is dominated by $\vec{a}_{s_o}$, where s_o is different from $k_o, k_1, \ldots, k_{r-1}$, and $x_{s_i} = \eta^i(x_{s_o})$ for $i = 1, 2, \ldots, r-1$, then $\vec{a}_{k_i}$ is dominated by $\vec{a}_{s_i}$ for $i = 1, 2, \ldots, r-1$, by repeatedly applying Lemma 7.6.3. Now let us prove that s_i is different from $k_o, k_1, \ldots, k_{r-1}$ for $i = 1, 2, \ldots, r-1$.

Assume $s_i = k_j$ for some j such that $1 \leq j \leq r-1$. Since η is a one-to-one mapping, we have

$$s_{i-1} = k_{j-1}$$

$$s_{i-2} = k_{j-2}$$

$$\vdots$$

$$s_o = \begin{cases} k_{j-1} & \text{if } j \geq i, \\ k_{r+j-i} & \text{if } j < i, \end{cases}$$

which contradicts that s_o is different from $k_o, k_1, \ldots, k_{r-1}$.

Q.E.D.

Lemma 7.6.5 Suppose the constraints matrix A has no dominating rows and r is the smallest positive integer such that $\eta^r(x_{k_o}) = x_{k_o}$ for a symmetric permutation η. If

(1) column $\vec{a}_{k_o}$ is dominated by $\vec{a}_{k_p}$ for some p such that $1 \leq p \leq r-1$,

(2) $x_{k_i} = \eta^i(x_{k_o})$ for $i = 1, 2, \ldots, r-1$,

then for each $i = 1, 2, \ldots, p-1$, $\vec{a}_{k_i} = \vec{a}_{k_{s_i}}$ for some s_i such that $p \leq s_i < r$.

Proof Since η^p is a symmetric permutation of (P), $\vec{a}_{k_p} = \vec{a}_{k_o}$ by Lemma 7.6.2. Since η^i is a symmetric permutation and $\vec{a}_{k_o}$ is dominated by $\vec{a}_{k_p}$, $\vec{a}_{k_i}$ is dominated by $\vec{a}_{k_{i+p}}$ for $i = 1, 2, \ldots, p-1$ by Lemma 7.6.3. Since $\eta^p(x_{k_i}) = \eta^{p+i}(x_{k_o}) = x_{k_{p+i}}$ and $\vec{a}_{k_i}$ is dominated by $\vec{a}_{k_{i+p}}$,

$$\vec{a}_{k_i} = \vec{a}_{k_{i+p}} \quad \text{for } i = 1, 2, \ldots, p-1. \tag{7.6.1}$$

by Lemma 7.6.2. If $2p \leq r$, then the lemma is proved by setting $s_i = p+1$. If $2p > r > p$, then there exists some $i \leq p-1$ such that $p+i \geq r$. For those i such that $p+i \geq r$, in the following of the proof, it will be shown that there exists some j such that

(1) $0 \leq j < i$,

(2) $\vec{a}_{k_{p+i}} = \vec{a}_{k_j}$,

(3) $p \leq j+p < r$.

Then the equalities $\vec{a}_{k_i} = \vec{a}_{k_{p+i}} = \vec{a}_{k_j} = \vec{a}_{k_{j+p}}$ hold (the last equality is obtained by replacing i in (7.6.1) by j), proving this lemma.

Let $j_i = p+i-r$. Since $x_{k_{p+i}} = \eta^{p+i}(x_{k_o}) = \eta^{p+i-r} \cdot \eta^r(x_{k_o}) = \eta^{p+i-r}(x_{k_o}) = \eta^{j_1}(x_{k_o}) = x_{k_j}$, we have $\vec{a}_{k_{p+i}} = \vec{a}_{k_{j_1}}$. Since $p < r$ and $p+i \geq r$, $0 \leq j_1 < i$. If $j_1+p < r$, then j_1 satisfies

$$(1) \quad 0 \leq j_1 < i, \tag{7.6.2}$$

$$(2) \quad \vec{a}_{k_{p+i}} = \vec{a}_{k_{j_1}}, \tag{7.6.3}$$

$$(3) \quad p \leq j_1+p < r, \tag{7.6.4}$$

i.e., j_1 is the j to be found. If $j_1+p \geq r$, then by letting

$j_2 = p+j_1-r$ and repeating the same argument applied to j_1, the following two formulas result.

$$\text{(1)} \quad 0 \le j_2 < j_1 < i. \tag{7.6.5}$$

$$\text{(2)} \quad \vec{a}_{k_{p+j_1}} = \vec{a}_{k_{j_2}} . \tag{7.6.6}$$

From (7.6.3), (7.6.1), and (7.6.6),

$$\vec{a}_{k_{p+i}} = \vec{a}_{k_{j_1}} = \vec{a}_{k_{p+j_1}} = \vec{a}_{k_{j_2}} . \tag{7.6.7}$$

If $j_2+p < r$, then from (7.6.5) and (7.6.7), j_2 is the j to be found. Since number i is a finite number, we must obtain some number j_q satisfying

$$\text{(1)} \quad 0 \le j_q < j_{q-1} < \cdots < j_1 < i,$$

$$\text{(2)} \quad \vec{a}_{k_{p+i}} = \vec{a}_{k_{j_1}} = \vec{a}_{k_{p+j_1}} = \cdots = \vec{a}_{k_{p+j_{q-1}}} = \vec{a}_{k_{j_q}} ,$$

$$\text{(3)} \quad p \le j_q+p < r,$$

if the argument applied to j_1 is repeatedly applied to $j_2, j_3, \cdots, j_{q-1}$.

Q.E.D.

Lemma 7.6.6 Suppose the constraint matrix A has no dominating row and r is the smallest positive integer such that $\eta^r(x_{k_o}) = x_{k_o}$ for a symmetric permutation η. If

(1) $x_{k_i} = \eta^i(x_{k_o})$ for $i = 1, 2, \ldots, r-1$,

(2) column $\vec{a}_{k_o}$ is dominated by $\vec{a}_{k_p}$ for some p such that $1 \le p \le r-1$,

then the problem (P2) obtained by deleting columns $\vec{a}_{k_o}, \vec{a}_{k_1}, \vec{a}_{k_2}, \ldots, \vec{a}_{k_{p-1}}$ from A (i.e., fixing variables $x_{k_o}, x_{k_1}, \ldots, x_{k_{p-1}}$ to 0) is symmetric under the permutation η' defined by

$$\eta' : \begin{cases} x_\ell \longrightarrow \eta(x_\ell), \text{ if } \ell \neq k_{r-1}, \\ x_{k_{r-1}} \longrightarrow x_p . \end{cases} \tag{7.6.8}$$

<u>Proof</u> The k_{p-1}-th column $\vec{b}_{k_{p-1}}$ of $\eta(A)$ is the k_p-th column $\vec{a}_{k_p}$ of A and the k_{r-1}-th column $\vec{b}_{k_{r-1}}$ of $\eta(A)$ is the k_o-th column $\vec{a}_{k_o}$ of A. Since η^p is symmetric and $x_{k_p} = \eta^p(x_{k_o})$, $\vec{a}_{k_o} = \vec{a}_{k_p}$ in A by Lemma 7.6.2, and consequently $\vec{b}_{k_{p-1}} = \vec{b}_{k_{r-1}}$ in $\eta(A)$.

Let $\hat{\eta}$ be a permutation defined by

$$\hat{\eta} : \begin{cases} x_\ell \longrightarrow \eta(x_\ell), \text{ if } 1 \neq k_{p-1}, k_{r-1}, \\ x_{k_{p-1}} \longrightarrow x_{k_o}, \\ x_{k_{r-1}} \longrightarrow x_{k_p} . \end{cases} \tag{7.6.9}$$

Then, from the definitions of $\hat{\eta}$ and $\hat{\eta}(A)$, $\hat{\eta}(A)$ is a matrix obtained from $\eta(A)$ by exchanging the k_{p-1}-th column $\vec{b}_{k_{p-1}}$ and the k_{r-1}-th column $\vec{b}_{k_{r-1}}$ in $\eta(A)$. Since $\vec{b}_{k_{p-1}} = \vec{b}_{k_{r-1}}$, matrix $\eta(A)$ is identical to $\hat{\eta}(A)$. Since η is symmetric, $\hat{\eta}$ is also symmetric by Theorem 7.4.1.

$$\begin{aligned} \text{Since } \quad & \hat{\eta}(x_{k_o}) = \eta(x_{k_o}) = x_{k_1}, \\ & \hat{\eta}^2(x_{k_o}) = \hat{\eta}\circ\hat{\eta}(x_{k_o}) = \hat{\eta}(x_{k_1}) = \eta(x_{k_1}) = x_{k_2}, \\ & \vdots \\ & \hat{\eta}^{p-1}(x_{k_o}) = \hat{\eta}\circ\hat{\eta}^{p-2}(x_{k_o}) = \hat{\eta}(x_{k_{p-2}}) = \eta(x_{k_{p-2}}) = x_{k_{p-1}}, \\ & \hat{\eta}^{p}(x_{k_o}) = \hat{\eta}\circ\hat{\eta}^{p-1}(x_{k_o}) = \hat{\eta}(x_{k_{p-1}}) = x_{k_o}, \end{aligned}$$

p is the smallest positive integer such that $\hat{\eta}^p(x_{k_o}) = x_{k_o}$. It is easy to see that the permutation η' of (7.6.8) is obtained from the symmetric permutation $\hat{\eta}$ of (7.6.9) by restricting it to (P2). By Theorem 7.5.3, η' is a symmetric permutation of (P2).

Q.E.D.

It will be proved in Theorem 7.6.7 that the following procedure DCDP (Dominated Columns Deletion Procedure) can be applied to reduce problem (P).

Procedure DCDP:

D1. Delete dominating rows from the constraint matrix A.

D2. Find a dominated column $\vec{a}_{k_o}$. If there is no column dominated by another, then the procedure terminates.

D3. Let $\eta(x_{k_o}) = x_{k_1}$, $\eta^2(x_{k_o}) = x_{k_2}$, ..., $\eta^{r-1}(x_{k_o}) = x_{k_{r-1}}$ and $\eta^r(x_{k_o}) = x_{k_o}$.

D3.1 If $\vec{a}_{k_o}$ is not dominated by any of $\vec{a}_{k_1}$, $\vec{a}_{k_2}$, ..., $\vec{a}_{k_{r-1}}$, then $\vec{a}_{k_o}$, $\vec{a}_{k_1}$, ..., $\vec{a}_{k_{r-1}}$ are deleted.

D3.2 If $\vec{a}_{k_o}$ is dominated by $\vec{a}_{k_p}$ for some p such that $1 \le p \le r-1$, then η is updated to $\hat{\eta}$, where $\hat{\eta}$ is defined by

$$\hat{\eta}: \begin{cases} x_\ell \longrightarrow \eta\,(x_\ell), & \ell \neq k_{r-1}, k_{p-1}, \\ x_{k_{r-1}} \longrightarrow x_p, & \\ x_{p-1} \longrightarrow x_{k_o}, & \end{cases}$$

and then $\vec{a}_{k_o}$, $\vec{a}_{k_1}$, ..., $\vec{a}_{k_{p-1}}$ are deleted.

D4. Update η to the permutation η' by restricting η (or $\hat{\eta}$ if η is updated in step D3.2) to the problem reduced at step D3.1. Go to step D1. ■

Theorem 7.6.7 Let (P) be a problem with symmetric permutation η. Then the procedure DCDP can be applied to reduce problem (P), and the last updated permutation η in the procedure DCDP is still a symmetric permutation of the reduced problem (P3).

Proof This theorem is proved by showing that

(1) columns $\vec{a}_{k_o}, \vec{a}_{k_1}, \ldots, \vec{a}_{k_{r-1}}$ in step D3.1 can be deleted without losing all the optimal solutions of problem (P),

(2) columns $\vec{a}_{k_o}, \vec{a}_{k_1}, \ldots, \vec{a}_{k_{p-1}}$ in step D3.2 can be deleted without losing all the optimal solutions of problem (P),

(3) symmetric permutation is preserved during steps D1 and D3 of the procedure.

From Lemma 7.6.1, symmetric permutation η is preserved when the procedure goes through step D1. In step D3, if $\vec{a}_{k_o}$ is dominated by some column other than $\vec{a}_{k_1}, \vec{a}_{k_2}, \ldots, \vec{a}_{k_{r-1}}$, then $\vec{a}_{k_i}$ is dominated by some column other than $\vec{a}_{k_1}, \vec{a}_{k_2}, \ldots, \vec{a}_{k_{r-1}}$, for $i = 1, 2, \ldots, r-1$, by Lemma 7.6.4. So $\vec{a}_{k_o}, \vec{a}_{k_1}, \ldots, \vec{a}_{k_{r-1}}$ can all be deleted without losing all optimal solutions of (P) since they are dominated columns. By Theorem 7.5.3, this reduced problem (the problem obtained by deleting $\vec{a}_{k_o}, \vec{a}_{k_1}, \ldots, \vec{a}_{k_{r-1}}$) is symmetric under the permutation η' obtained by restricting η to the reduced problem. If $\vec{a}_{k_o}$ is dominated by $\vec{a}_{k_p}$ for some p such that $1 \leq p \leq r-1$, then, for each $i = 1, 2, \ldots, p-1$, we have $\vec{a}_{k_i} = \vec{a}_{k_{s_i}}$ for some s_i such that $p \leq s_i < r$ by Lemma 7.6.5. So $\vec{a}_{k_o}, \vec{a}_{k_2}, \ldots, \vec{a}_{k_p}$ can all be deleted without losing all

optimal solutions of (P) since they are columns dominated by others (i.e., $\vec{a}_{k_i} = \vec{a}_{k_{s_i}}$).

By Lemma 7.6.6, the reduced problem (the problem obtained by deleting $\vec{a}_{k_o}$, $\vec{a}_{k_1}$, ..., $\vec{a}_{k_{p-1}}$) is symmetric under the permutation η' obtained by restricting $\hat{\eta}$ to the reduced problem (η' is the one defined in Lemma 7.6.6). Thus the symmetric permutation η is preserved during step D3.

Q.E.D.

From the above theorem, if the DCDP procedure is applied to a given problem (P) with a symmetric permutation η, then the reduced problem (P3) will have no dominating row and dominated column in its reduced constraint matrix. Also, (P3) is symmetric under a symmetric permutation η' which is obtained from η by the corresponding reduction.

<u>Lemma 7.6.8</u> Let η be a symmetric permutation of problem (P). If

(1) $\vec{a}_{k_o}$ is an essential column of the constraint matrix A,

(2) $x_{k_i} = \eta^i (x_{k_o})$ for i = 1, 2, ..., r-1, where $\eta^r (x_{k_o}) = x_{k_o}$,

then columns $\vec{a}_{k_o}$, $\vec{a}_{k_1}$, ..., $\vec{a}_{k_{r-1}}$ are all essential.

<u>Proof</u> This lemma is proved by showing that if $x_t = \rho (x_{k_o})$ for some symmetric permutation ρ, then $\vec{a}_t$ is also essential. Then since $\eta, \eta^2, \ldots, \eta^{r-1}$ are symmetric permutations, $\vec{a}_{k_o}$, $\vec{a}_{k_1}$, $\vec{a}_{k_2}$, ..., $\vec{a}_{k_{r-1}}$ are all essential.

Since $\vec{a}_{k_o}$ is essential, there exists some $\vec{r}_i = (a_{i1}, a_{i2}, \ldots, a_{in})$ of A such that $a_{ik_o} = 1$ and $a_{ij} = 0$ for $j \neq k_o$. Since ρ is a symmetric permutation of (P), $\vec{r}_i$ dominates some row $\vec{r}'_\ell = (b_{\ell 1}, b_{\ell 2}, \ldots, b_{\ell n})$ in $\rho(A)$, by Theorem 7.4.1. Since $a_{ik_o} = 1$ is the only non-zero element in $\vec{r}_i$, $b_{\ell j} = 0$ for all $j \neq k_o$ and $b_{\ell k_o} = 1$. Since $x_t = \rho(x_{k_o})$, the k_o-th column $\vec{b}_{k_o}$ of $\rho(A)$ is the t-th column $\vec{a}_t$ of A, by the definition of $\rho(A)$. So $a_{\ell t} = b_{\ell k_o} = 1$ and $a_{\ell j} = 0$ for all $j \neq t$, i.e., $\vec{a}_t$ is also essential.

Q.E.D.

Lemma 7.6.9 Let η be a symmetric permutation of problem (P). If $x_{k_i} = \eta^i(x_{k_o})$ for $i = 1, 2, \ldots, r-1$, where $x_{k_o} = \eta^r(x_{k_o})$, then problem (P4) obtained by fixing variables $x_{k_o}, x_{k_1}, \ldots, x_{k_{r-1}}$ to 1 and deleting all rows covered by columns $\vec{a}_{k_o}, \vec{a}_{k_1}, \ldots, \vec{a}_{k_{r-1}}$, is symmetric under the permutation η' obtained by restricting η to the problem (P4).

Proof Let A' be the constraint matrix of (P4). First, we will show that each row $\vec{r}'_i = (a'_{i1}\ a'_{i2}, \ldots, a'_{in'})$ of A' dominates some row of $\eta'(A')$.

Let $\vec{r}_j = (a_{j1}, a_{j2}, \ldots, a_{jn})$ be the row in A such that $\vec{r}'_i$ is obtained by deleting the k_o-th, k_1-th, ..., k_{r-1}-th element from $\vec{r}_j$. Since $\vec{r}'_i$ is a row of A', $a_{jk_o}, a_{jk_1}, \ldots, a_{jk_{r-1}}$ must be 0 (otherwise, $\vec{r}'_i$ will not be a row of A'). Since η is symmetric, $\vec{r}_j$ dominates some row $\vec{v}_s = (b_{s1}, b_{s2}, \ldots, b_{sn})$ of $\eta(A)$, by Theorem

7.4.1. Since $a_{jk_o}, a_{jk_1}, \ldots, a_{jk_{r-1}}$ are 0, $b_{sk_o}, b_{sk_1}, \ldots, b_{sk_{r-1}}$ are also 0. Let $\vec{v}'_s$ be the row obtained by deleting $b_{sk_o}, b_{sk_1}, \ldots, b_{sk_{r-1}}$ from $\vec{v}_s$. Then $\vec{r}'_i$ dominates $\vec{v}'_s$.

Now let us show that $\vec{v}'_s$ is a row of $\eta'(a')$.

Since η' is the permutation obtained from η by restricting it to (P4), $\eta'(A')$ is obtained from $\eta(A)$ by deleting the k_o-th, k_1-th,$\cdots$, k_{r-1}-th columns and deleting all the rows covered by the k_o-th, k_1-th,$\cdots$, k_{r-1}-th columns. Since $b_{sk_o}, b_{sk_1}, \ldots, b_{sk_{r-1}}$ are 0, $\vec{v}'_s$ is a row of $\eta'(A')$.

Thus, $\vec{r}'_i$ of A' dominates $\vec{v}'_s$ of $\eta'(A')$. By Theorem 7.4.1 (P4) is symmetric under η'.

Q.E.D.

It will be proved in Theorem 7.6.10 that the following procedure <u>ECFP</u> (Essential Columns Finding Procedure) can be applied to reduce problem (P).

<u>Procedure ECFP</u>

<u>E1</u>. Find an essential column $\vec{a}_{k_o}$ of the constraint matrix. If no essential column exists, then the procedure terminates.

<u>E2</u>. Let $\eta(x_{k_o}) = x_{k_1}$, $\eta^2(x_{k_o}) = x_{k_2}$, $\ldots$, $\eta^{r-1}(x_{k_o}) = x_{k_{r-1}}$, $\eta^r(s_{k_o}) = x_{k_o}$. Fix $x_{k_o}, x_{k_1}, \ldots, x_{k_{r-1}}$ to 1 and delete all rows covered by $\vec{a}_{k_o}, \vec{a}_{k_1}, \ldots, \vec{a}_{k_{r-1}}$.

<u>E3</u>. Update η to the permutation η' obtained by restricting η to the problem reduced at step E2, and go to step E1. ∎

Theorem 7.6.10 Let (P) be a problem with a symmetric permutation η. Then the procedure ECFP can be applied to reduce problem (P), and the last updated permutation η in the procedure ECFP is a symmetric permutation of the reduced problem (P5).

Proof From Lemma 7.6.8, if $\vec{a}_{k_o}$ is essential, then $\vec{a}_{k_1}$, $\vec{a}_{k_2}$, ..., $\vec{a}_{k_{r-1}}$ are all essential. So x_{k_o}, x_{k_1}, ..., $x_{k_{r-1}}$ can all be fixed to 1 in step E2 without losing any optimal solution of (P). From Lemma 7.6.9, the reduced problem (problem obtained by fixing x_{k_o}, x_{k_1}, ..., $x_{k_{r-1}}$ to 1) is symmetric under the permutation η' obtained by restricting η to this reduced problem.

Q.E.D.

If the ECFP procedure is applied to a given problem (P) with a symmetric permutation η, then the reduced problem (P5) will not have any essential column in its reduced constraint matrix.

From Theorems 7.6.7 and 7.6.10, if the procedures DCDP and ECFP are repeatedly applied to the problem (P) with a symmetric permutation η, then the reduced problem (P6), where none of the three reduction operations described in the beginning of Section 7.6 can be applied is still a problem with symmetric permutation η', which is obtained from η by updating it as described in Theorems 7.6.7 and 7.6.10.

7.7 Preservation Of Symmetric Permutations With Different Generators

In this section a problem with symmetric permutations of more than one generator is considered. Throughout this section it is

assumed that $\eta_1, \eta_2, \ldots, \eta_h$ are different generators of symmetric permutations of the problem (P). It is also assumed that r_i is the smallest positive integer such that $\eta_i^{r_i}$ is the identity permutation for i = 1, 2, ..., h.

An example of a problem with symmetric permutations of more than one generator has already been shown in Section 7.2. In the following another example is shown. This example is used for the illustration later in this section.

Example 7.7.1 Consider a problem with a constraint matrix

$$A = \left[\begin{array}{c|c|c} B & & C \\ \hline C & B & \\ \hline & C & B \end{array}\right], \tag{7.7.1}$$

where

$$B = \left[\begin{array}{c|c|c} D & & E \\ \hline E & D & \\ \hline & E & D \end{array}\right], \quad C = \left[\begin{array}{c|c|c} F & & G \\ \hline G & F & \\ \hline & G & F \end{array}\right],$$

D, E, F, G are arbitrary $n \times n$ zero-one matrices, and the blank areas show all 0's.

Define a permutation η_1 on $X = \{x_1, x_2, \ldots, x_{9n}\}$

as

$$\eta_1 : \begin{cases} x_i \longrightarrow x_{i+6n} & \text{if } 1 \le i \le 3n, \\ \\ x_i \longrightarrow x_{i-3n} & \text{if } 3n < i \le 9n. \end{cases} \tag{7.7.2}$$

Then, from the definition of $\eta_1(A)$,

$$\eta_1(A) = \left[\begin{array}{c|c|c} C & B & \\ \hline & C & B \\ \hline B & & C \end{array}\right] . \tag{7.7.3}$$

Comparing $\eta_1(A)$ with A, it is easy to see that each row of A dominates some row of $\eta_1(A)$. So η_1 is a symmetric permutation of this problem, by Theorem 7.4.1.

Define another permutation η_2 on $X = \{x_1, x_2, \ldots, x_{9n}\}$ as

$$\eta_2 : \begin{cases} x_i \longrightarrow x_{i+2n}, & \text{if } 1 \le i \le n,\ 3n < i \le 4n,\ 6n < i \le 7n, \\ \\ x_i \longrightarrow x_{i-n}, & \text{if } n < i \le 3n,\ 4n < i \le 6n,\ 7n < i \le 9n. \end{cases} \tag{7.7.4}$$

By the definition of $\eta_2(A)$,

$$\eta_2(A) = \left[\begin{array}{ccc|ccc|ccc}
E & D & & & & & G & F & \\
 & E & D & & & & & G & F \\
D & & E & & & & F & & G \\
\hline
G & F & & E & D & & & & \\
 & G & F & & E & D & & & \\
F & & G & D & & E & & & \\
\hline
 & & & G & F & & E & D & \\
 & & & & G & F & & E & D \\
 & & & F & & G & D & & E
\end{array}\right] \tag{7.7.5}$$

Writing A explicitly in D, E, F, G, we have

$$A = \left[\begin{array}{ccc|ccc|ccc}
D & & E & & & & F & & G \\
E & D & & & & & G & F & \\
 & E & D & & & & & G & F \\
\hline
F & & G & D & & E & & & \\
G & F & & E & D & & & & \\
 & G & F & & E & D & & & \\
\hline
 & & & F & & G & D & & E \\
 & & & G & F & & E & D & \\
 & & & & G & F & & E & D
\end{array}\right] \tag{7.7.6}$$

Comparing A in (7.7.6) with $\eta_2(A)$ in (7.7.5), it is easy to see that each row of A dominates some row of $\eta_2(A)$. So η_2 is also a symmetric permutation of this problem, by Theorem 7.4.1.

From the definition of η_1 and η_2, $\eta_1 \neq \eta_2^i$ for any positive integer i and $\eta_2 \neq \eta_1^j$ for any positive integer j. Thus, the problem with the constraint matrix in the form (7.7.1) is a problem with symmetric permutations of two different generators.

■

For a given positive integer ℓ, define $H^{\ell}_{\eta_1\eta_2, \ldots, \eta_h}(x_{k_o})$ to be the set of variables such that each variable x_v in it can be expressed as

$$x_v = \eta_{j_k} \circ \eta_{j_{k-1}} \circ \ldots \circ \eta_{j_1} (x_{k_o})$$

for some $k \leq \ell$ and some $\eta_{j_1}, \eta_{j_2}, \ldots, \eta_{j_k} \in \{\eta_1, \eta_2, \ldots, \eta_h, I$ (the identity)$\}$. Note that some of $\eta_{j_1}, \eta_{j_2}, \ldots, \eta_{j_k}$ in the above definition may be the same, and the variable x_{k_o} is a variable in $H^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$, since $x_{k_o} = I\,(x_{k_o})$. As a special case, $H^{o}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ is defined to be the set $\{x_{k_o}\}$.

Example 7.7.2 From the above definition, if η_1 and η_2 are defined as in (7.7.2) and (7.7.4), then

$$H^1_{\eta_1}(x_1) = \{x_1, x_{6n+1}\},$$

$$H^1_{\eta_2}(x_1) = \{x_1, x_{2n+1}\},$$

$$H^1_{\eta_1\eta_2}(x_1) = \{x_1, x_{6n+1}, x_{2n+1}\},$$

$$H^2_{\eta_1}(x_1) = \{x_1, x_{6n+1}, x_{3n+1}\},$$

$$H^2_{\eta_2}(x_1) = \{x_1, x_{2n+1}, x_{n+1}\},$$

$$H^2_{\eta_1\eta_2}(x_1) = \{x_1, x_{6n+1}, x_{2n+1}, x_{3n+1}, x_{n+1}, x_{8n+1}\},$$

$$H^3_{\eta_1}(x_1) = \{x_1, x_{6n+1}, x_{3n+1}\},$$

$$H^3_{\eta_2}(x_1) = \{x_1, x_{2n+1}, x_{n+1}\},$$

$$H^3_{\eta_1\eta_2}(x_1) = \{x_1, x_{6n+1}, x_{2n+1}, x_{3n+1}, x_{n+1}, x_{8n+1}, x_{5n+1}, x_{7n+1}\},$$

$$H^4_{\eta_1}(x_1) = H^3_{\eta_1}(x_1),$$

$$H^4_{\eta_2}(x_1) = H^3_{\eta_2}(x_1),$$

$$H^4_{\eta_1\eta_2}(x_1) = \{x_1, x_{6n+1}, x_{2n+1}, x_{3n+1}, x_{n+1}, x_{8n+1}, x_{5n+1}, x_{7n+1}, x_{4n+1}\},$$

$$H^5_{\eta_1\eta_2}(x_1) = H^4_{\eta_1\eta_2}(x_1).$$

∎

For a given positive integer ℓ, define

$$D^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}) = H^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}) - H^{\ell-1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}). \tag{7.7.7}$$

As a special case, define $D^{o}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}) = \{x_{k_o}\}$. From (7.7.7),

$$H^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}) = H^{\ell-1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}) + D^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}). \tag{7.7.8}$$

Theorem 7.7.1 If $\ell > 0$, then each variable in $D^{\ell-1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ is mapped from some variable in $D^{\ell-1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ by some symmetric

permutation η_i where $1 \leq i \leq h$.

Proof Each variable x_t in $D^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ can be expressed as

$\eta_{j_\ell} \circ \eta_{j_{\ell-1}} \circ \cdots \circ \eta_{j_1}(x_{k_o})$ for some $j_1, j_2, \ldots, j_\ell \in \{1, 2, \ldots, h\}$,

by (7.7.7). Let $x_v = \eta_{j_{\ell-1}} \circ \cdots \circ \eta_{j_1}(x_{k_o})$. Then we have to show

that x_v is in $D^{\ell-1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$. If x_v is not in $D^{\ell-1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$,

then x_v must be in $H^{\ell-2}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ by (7.7.7), i.e., x_v can be expressed as $\eta_{j'_k} \circ \eta_{j'_{k-1}} \circ \cdots \circ \eta_{j'_1}(x_{k_o})$ for some $k \leq \ell - 2$ and

some $j'_1, j'_2, \ldots, j'_k$ in $\{1, 2, \ldots, h\}$. Then

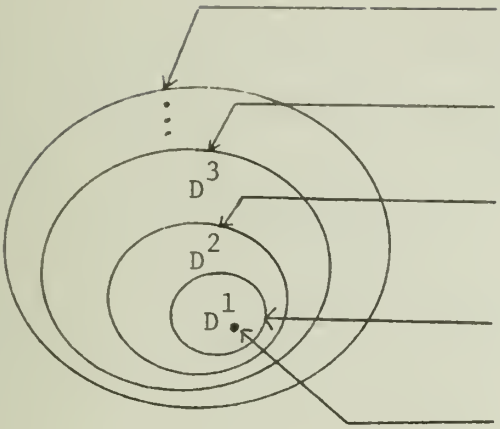
$$
\begin{aligned}
x_t &= \eta_{j_\ell} \circ \eta_{j_{\ell-1}} \circ \cdots \circ \eta_{j_1}(x_{k_o}), \\
&= \eta_{j_\ell}(x_{k_o}) \qquad\qquad (7.7.9) \\
&= \eta_{j_\ell}\, \eta_{j'_k} \circ \cdots \circ \eta_{j'_1}(x_{k_o})
\end{aligned}
$$

where $k \leq \ell - 2$. Equalities (7.7.9) shows that $x_t \in H^{\ell-1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$,

contradicting the assumption that $x_t \in D^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}) = H^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}) - H^{\ell-1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$.

Q.E.D.

The relation between $H^{i}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ and $D^{i}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ for each i is shown in Figure 7.7.1.



$H^{\infty}_{\eta_1\eta_2\,\cdots\,\eta_h}(x_{k_o}) = G_{\eta_1\eta_2\,\cdots\,\eta_h}(x_{k_o})$

$H^{3}_{\eta_1\eta_2\,\cdots\,\eta_h}(x_{k_o})$

$H^{2}_{\eta_1\eta_2\,\cdots\,\eta_h}(x_{k_o})$

$H^{1}_{\eta_1\eta_2\,\cdots\,\eta_h}(x_{k_o})$

$H^{o}_{\eta_1\eta_2\,\cdots\,\eta_h}(x_{k_o}) = \{x_{k_o}\}$

Figure 7.7.1 The relation between $H^{i}_{\eta_1\eta_2\,\cdots\,\eta_h}(x_{k_o})$ and $D^{i}_{\eta_1\eta_2\,\cdots\,\eta_h}(x_{k_o})$.

<u>Example 7.7.3</u> If η_1 and η_2 are defined in (7.7.2) and (7.7.4), then

$$D^1_{\eta_1}(x_1) = \{x_{6n+1}\},$$

$$D^1_{\eta_1}(x_1) = \{x_{2n+1}\},$$

$$D^1_{\eta_1\eta_2}(x_1) = \{x_{6n+1}, x_{2n+1}\},$$

$$D^2_{\eta_1}(x_1) = \{x_{3n+1}\},$$

$$D^2_{\eta_2}(x_1) = \{x_{n+1}\},$$

$$D^2_{\eta_1\eta_2}(x_1) = \{x_{3n+1}, x_{n+1}, x_{8n+1}\},$$

$$D^3_{\eta_1}(x_1) = \text{empty},$$

$$D^3_{\eta_2}(x_1) = \text{empty},$$

$$D^3_{\eta_1\eta_2}(x_1) = \{x_{5n+1}, x_{7n+1}\},$$

$D^{4}_{\eta_1\eta_2}(x_1) = \{x_{4n+1}\}$,

$D^{5}_{\eta_1\eta_2}(x_1) =$ empty.

■

Denote $H^{\infty}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ by $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$. From this definition of $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$, it is easy to see that $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ is the set of variables such that each variable x_v in it can be expressed as

$$x_v = \eta_{j_k}^{p_k} \circ \eta_{j_{k-1}}^{p_{k-1}} \circ \cdots \circ \eta_{j_1}^{p_1}(x_{k_o}),$$

for some positive integers k, $p_1, p_2, \ldots, p_k$ and some $\eta_{j_1}, \eta_{j_2}, \ldots, \eta_{j_k} \varepsilon \{\eta_1, \eta_2, \ldots, \eta_h, I \text{ (the identity permutation)}\}$.

Since $\eta_{j_k}^{p_k} \circ \eta_{j_{k-1}}^{p_{k-1}} \circ \cdots \circ \eta_{j_1}^{p_1}$ is a symmetric permutation of (P) for any positive integers k, $p_1, p_2, \ldots, p_k$ and any set of subscripts $j_1, j_2, \ldots, j_k$, each variable in $G_{\eta_1\eta_2 \cdots \eta_f}(x_{k_o})$ can be fixed to 0 without losing a better feasible solution (a solution better than the best one obtained so far) in the subproblem with x_{k_o} fixed to 0, by Theorem 7.1.1, if the subproblem with x_{k_o} fixed to 1 has already been enumerated.

Theorem 7.7.2 Let $\eta_1, \eta_2, \ldots, \eta_h$ be symmetric permutations of the problem (P) and $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}) = \{x_{k_o}, x_{k_1}, \ldots, x_{k_q}\}$. Then after the subproblem with x_{k_o} fixed to 1 has been enumerated, variables

$x_{k_1}, x_{k_2}, \ldots, x_{k_q}$ can be fixed to 0 without losing a better feasible solution in the subproblem with $x_{k_o} = 0$.

<u>Proof</u> Since (1) the subproblem with variables $x_{k_o}, x_{k_1}, \ldots, x_{k_{i-1}}$ all fixed to 0 is a subproblem of the subproblem with x_{k_o} fixed to 0, and (2) x_{k_i} can be fixed to 0 without losing a better feasible solution in the subproblem with x_{k_o} fixed to 0 by Theorem 7.1.1, x_{k_i} can be further fixed to 0 without losing a better feasible solution in the subproblem with $x_{k_o}, x_{k_1}, \ldots, x_{k_{i-1}}$ fixed to 0 if the subproblem with x_{k_o} fixed to 1 has been enumerated. The above argument can be applied for $i = 1, 2, \ldots, q$. Thus, after the subproblem with x_{k_o} fixed to 1 has been enumerated, $x_{k_o}, x_{k_1}, \ldots, x_{k_q}$ can be fixed to 0 without losing a better feasible solution of (P) by repeatedly fixing variable x_{k_i} to 0 in the subproblem with $x_{k_o}, x_{k_1}, \ldots, x_{k_{i-1}}$ fixed to 0 for $i = 1, 2, \ldots, q$.

Q.E.D.

Let (P7) be the subproblem obtained from (P) by fixing variables $x_{k_o}, x_{k_1}, \ldots, x_{k_q}$ to 0 and let $X' = X - G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$.

<u>Theorem 7.7.3</u> If x_ℓ is not a variable in $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$, then $\eta_i(x_\ell)$ is not a variable in $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ for $i = 1, 2, \ldots, h$.

<u>Proof</u> Let r_i be the smallest positive integer such that $\eta_i^{r_i}$ is the

identity permutation. If $\eta_i(x_\ell)$ is a variable in $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$, then $\eta_i(x_\ell) = \eta_{j_k}^{p_k} \circ \eta_{j_{k-1}}^{p_{k-1}} \circ \ldots \circ \eta_{j_1}^{p_1}(x_{k_o})$ for some positive integers k, $p_1, p_2, \ldots, p_k$ and some $j_1, j_2, \ldots, j_k$. Obviously $\eta_i^{r_i-1} \circ \eta_i(x_\ell) = \eta_i^{r_i}(x_\ell) = x_\ell$ holds and this can be rewritten as

$$x_\ell = \eta_i^{r_i-1} \circ \eta_{j_k}^{p_k} \circ \eta_{j_{k-1}}^{p_{k-1}} \circ \ldots \circ \eta_{j_1}^{p_1}(x_{k_o}),$$

which shows that x_ℓ is a variable in $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$. This contradicts that x_ℓ is not a variable in $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$.

Q.E.D.

From Theorem 7.7.3, for each $i = 1, 2, \ldots, h$, a permutation η_i' on $X' = X - G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ can be defined as

$$\eta_i'(x_\ell) = \eta_i(x_\ell) \text{ for all } x_\ell \text{ in } X'. \qquad (7.7.10)$$

$\eta_1', \eta_2', \ldots, \eta_h'$ are said to be <u>obtained from</u> $\eta_1, \eta_2, \ldots, \eta_h$ <u>by restricting</u> them to (P7). In the following Theorem 7.7.4, $\eta_1', \eta_2' \ldots, \eta_h'$ are proved to be symmetric permutations of (P7).

<u>Theorem 7.7.4</u> If (P7)(or (P8))is the problem obtained from (P) by fixing all variables in $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ to 0 (or to 1), then permutations $\eta_1', \eta_2', \ldots, \eta_h'$, which are obtained by restricting $\eta_1, \eta_2, \ldots, \eta_h$ to (P7) (or (P8)), are symmetric permutations of (P7) (or P8)).

<u>Proof</u> Let η_t be one of $\eta_1, \eta_2, \ldots, \eta_h$. In the following, we will prove that η_t' is a symmetric permutation of (P7) (or (P8)). Then, $\eta_1', \eta_2', \ldots, \eta_h'$ will be symmetric permutation of (P7) (or (P8)).

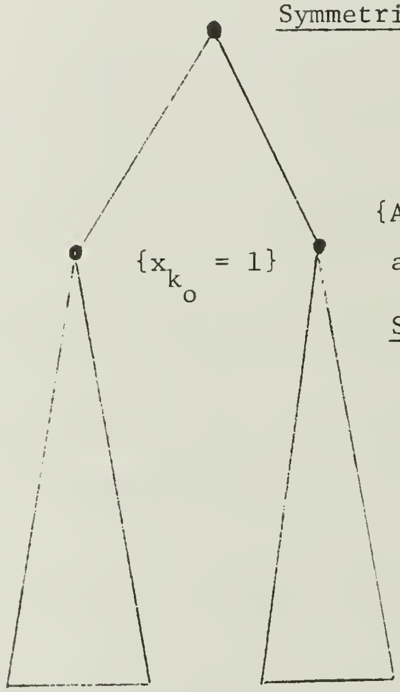
Let x_{i_o} be a variable in $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$. Then $x_{i_o} = \eta_{j_k}^{p_k} \circ \eta_{j_{k-1}}^{p_{k-1}} \circ \cdots \circ \eta_{j_1}^{p_1}(x_{k_o})$ for some positive integers k, $p_1, p_2, \ldots, p_k$ and some $j_1, j_2, \ldots, j_k \in \{1, 2, \ldots, h\}$. Since $\eta_t^{\ell}(x_{i_o}) = \eta_t^{\ell} \circ \eta_{j_k}^{p_k} \circ \eta_{j_{k-1}}^{p_{k-1}} \circ \cdots \circ \eta_{j_1}^{p_1}(x_{k_o})$, $\eta_t^{\ell}(x_{k_o})$ is a variable in $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ for $\ell = 1, 2, \ldots, r_t - 1$, where $\eta_t^{r_t}$ is the identity permutation. If variables $x_{i_o}, \eta_t(x_{i_o}), \ldots, \eta_t^{r_{t-1}}(x_{i_o})$ are fixed to 0 (or to 1), then the reduced (P^*) is symmetric under the permutation η_t^* obtained from η_t by restricting it to (P^*), by Theorem 7.5.3 (or by Lemma 7.6.9). If all variables in $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ are already fixed to 0 (or to 1), then (P7) = (P^*)(or (P8) = (P^*)) and (P7) (or (P8)) is symmetric under $\eta_t' = \eta_t^*$. If there exist some variables in $G_{\eta_1\eta_2 \cdots \eta_f}(x_{k_o})$ not fixed yet, let x_{i_1} be one of them. By the same argument applied to x_{i_o}, another reduced problem (P^{**}) with more variables in $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ fixed to 0 (or to 1) than before is obtained and this reduced problem (P^{**}) is symmetric under the permutation η_t^{**} obtained by restricting η_t^* to (P^{**}).

Repeating the above process, problem (P7) (or P(8)) will be obtained,since the number of variables in $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ is finite. Problem (P7) (or problem (P8)) is symmetric under η_t' since the problem

obtained after each process discussed above is symmetric under the permutation obtained by restricting η_t to it.

Q.E.D.

All the discussions in this section are illustrated in Figure 7.7.2.



Symmetric permutations: $\eta_1, \eta_2, \ldots, \eta_h$

{All variables in $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ are fixed to 0}

Symmetric permutations:

$\eta_1', \eta_2', \ldots, \eta_h'$ (η_i' is obtained from η_i by restricting it to this subproblem for each i).

Figure 7.7.2 Illustration of a problem with more than one symmetric generator when program backtracks.

The following is an efficient procedure to find $G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ for a given x_{k_o}, based on Theorem 7.7.1.

Procedure GF ($G_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ Finding Procedure):

F1. $H \leftarrow \{x_{k_o}\}$, $D \leftarrow \{x_{k_o}\}$, $D1 \leftarrow$ empty.

F2. For each i = 1, 2, ..., h and each x_v in D, if $\eta_i(x_v)$ is not a variable in H or D, then store $\eta_i(x_v)$ in D1.

F3. H ← (Union of H and D), D ← D1, D1 ← empty.

F4. If D is empty, then the procedure terminates.

Otherwise go to step F2. ■

By applying the above procedure to a given variable x_{k_o}, $G_{\eta_1 \eta_2 \cdots \eta_h}(x_{k_o})$ is obtained as the set H in the above procedure.

In the following we shall show that $\eta_1, \eta_2, \ldots, \eta_h$ are preserved during the three reduction operations stated in Section 7.6.

Lemma 7.7.5 Suppose there is no dominating row in the constraint matrix A of (P) and $\eta_1, \eta_2, \ldots, \eta_h$ are symmetric permutations of (P). If column $\vec{a}_{k_o}$ is dominated by some other column, then column $\vec{a}_{k_i}$ is dominated by some other column for every $x_{k_i} \in G_{\eta_1 \eta_2 \cdots \eta_h}(x_{k_o}) = \{x_{k_o}, x_{k_1}, \ldots, x_{k_q}\}$.

Proof Since $x_{k_i} \in G_{\eta_1 \eta_2 \cdots \eta_h}(x_{k_o})$, $x_{k_i} = \eta_{j_k}^{p_k} \circ \eta_{j_{k-1}}^{p_{k-1}} \circ \cdots \circ \eta_{j_1}^{p_1}(x_{k_o})$ for some positive integers k, $p_1, p_2, \ldots, p_k$ and some $j_1, j_2, \ldots, j_k \in \{1, 2, \ldots, h\}$. Since $\eta_{j_k}^{p_k} \circ \eta_{j_{k-1}}^{p_{k-1}} \circ \cdots \circ \eta_{j_1}^{p_1}$ is symmetric and $\vec{a}_{k_o}$ is dominated by some other column say $\vec{a}_{s_o}$, then $\vec{a}_{k_i}$ is dominated by $\vec{a}_{s_i}$, where $x_{s_i} = \eta_{j_k}^{p_k} \circ \eta_{j_{k-1}}^{p_{k-1}} \circ \cdots \circ \eta_{j_1}^{p_1}(x_{s_o})$, by Lemma 7.6.3.

Q.E.D.

Lemma 7.7.6 Suppose there is no dominating row in the constraint matrix A of (P) and $\eta_1, \eta_2, \ldots, \eta_h$ are symmetric permutations of (P) such that column $\vec{a}_s$, does not dominate column $\vec{a}_t$, for every pair of

variables $x_{s'}$ and $x_{t'}$ in $H^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$. If there exists a pair variables x_s and x_t in $H^{\ell+1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ such that $\vec{a}_s$ dominates $\vec{a}_t$, then

(1) $\vec{a}_s = \vec{a}_t$,

(2) one of x_s and x_t, say x_s, must be a variable in $D^{\ell+1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$, ($x_t$ may or may not be in $D^{\ell+1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$),

(3) there exist symmetric permutations $\eta_1^*, \eta_2^*, \ldots, \eta_h^*$ of (P) such that

(a) each η_i^* is obtained from η_i by the following modification: If there exists some variable x_v in $D^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ such that $\eta_i(x_v) = x_s$ then η_i^* is defined as

$$\eta_i^*: \begin{cases} x_d \longrightarrow \eta_i(x_d), \text{ if } x_d \neq x_u, x_v, \\ x_u \longrightarrow x_s, \\ x_v \longrightarrow x_t, \end{cases} \tag{7.7.10}$$

where x_u is the variable such that $\eta_i(x_u) = x_t$.

Otherwise $\eta_i^* = \eta_i$.

(b) $$H^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}) = H^{\ell}_{\eta_1^*\eta_2^* \cdots \eta_h^*}(x_{k_o}) \tag{7.7.11}$$

(c) $$D^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}) = D^{\ell}_{\eta_1^*\eta_2^* \cdots \eta_h^*}(x_{k_o}) \tag{7.7.12}$$

(d) $x_s \notin D^{\ell+1}_{\eta_1^*\eta_2^* \cdots \eta_h^*}(x_{k_o})$ (7.7.13)

(e) $D^{\ell+1}_{\eta_1^*\eta_2^* \cdots \eta_h^*}(x_{k_o}) \subsetneq D^{\ell+1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ (7.7.14)

<u>Proof</u> Since x_s and x_t are variables in $H^{\ell+1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$, for some $x_s = \eta_{j_k} \circ \eta_{j_{k-1}} \circ \eta_{j_1}(x_{k_o})$ and $x_t = \eta_{j'_{k'}} \circ \eta_{j'_{k'-1}} \circ \eta_{j'_1}(x_{k_o})$ for some $j_1, j_2, \ldots, j_k, j'_1, j'_2, \ldots, j'_{k'} \varepsilon \{1, 2, \ldots, h\}$, where $k \le \ell + 1$, and $k' \le \ell + 1$. Since $\eta_{j_k} \circ \eta_{j_{k-1}} \circ \ldots \circ \eta_{j_1}$ and $\eta_{j'_{k'}} \circ \eta_{j'_{k'-1}} \circ \ldots \circ \eta_{j'_1}$ are symmetric, the numbers of non-zero elements in each of $\vec{a}_{k_o}$, $\vec{a}_s$, and $\vec{a}_t$ are the same, by Lemma 7.6.2. Since $\vec{a}_s$ dominates $\vec{a}_t$ and the numbers of non-zero elements in $\vec{a}_s$ and $\vec{a}_t$ are the same,

$$\vec{a}_s = \vec{a}_t \qquad (7.7.15)$$

Thus, (1) is proved.

Since $\vec{a}_{s'}$ does not dominate $\vec{a}_{t'}$ for any pair of variables $x_{s'}$ and $x_{t'}$ in $H^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$, at least one of x_s and x_t must be a member of $D^{\ell+1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$. (This is because $D^{\ell+1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}) = H^{\ell+1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}) - H^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$). Thus, (2) is proved.

Let x_s be a member of $D^{\ell+1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$. Since x_s is in $D^{\ell+1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$, the following equations are true:

(i) $k = \ell + 1$, where k is the subscript of j_k such that $x_s = \eta_{j_k} \circ \eta_{j_{k-1}} \circ \ldots \circ \eta_{j_1}$.

(ii) $x_v = \eta_{j_{k-1}} \circ \eta_{j_{k-2}} \circ \dots \circ \eta_{j_1}(x_{k_o})$ is a member of

$$D^{\ell}_{\eta_1\eta_2 \dots \eta_h}(x_k) \text{ (by Theorem 7.7.1).} \tag{7.7.16}$$

(iii) $x_s = \eta_{j_k} \circ \eta_{j_{k-1}} \circ \dots \circ \eta_{j_1}(x_{k_o}) = \eta_{j_k}(x_v)$.

In the above equation (iii), η_{j_k} is one of $\eta_1, \eta_2, \dots, \eta_h$. Let it be η_i and let x_u be the variable such that $\eta_i(x_u) = x_t$. Define η_i^* as

$$\eta_i^* : \begin{cases} x_d \longrightarrow \eta_i(x_d), \text{ if } x_d \neq x_u, x_v \\ x_u \longrightarrow x_s, \\ x_v \longrightarrow x_t. \end{cases} \tag{7.7.17}$$

Since $\vec{a}_s = \vec{a}_t$, matrix $\eta_i(A)$ is identical to matrix $\eta_i^*(A)$ by the definitions of η_i and η_i^*. Thus,

$$\eta_i^* \text{ is a symmetric permutation of (P),} \tag{7.7.18}$$

by Theorem 7.4.1.

Next, let us show that

(A) x_u is not a member of $H^{\ell}_{\eta_1\eta_2 \dots \eta_h}(x_{k_o})$, (7.7.19)

(B) $H^{\ell}_{\eta_1\eta_2 \dots \eta_{i-1}\eta_i^*\eta_{i+1} \dots \eta_h}(x_{k_o}) =$

$$H^{\ell}_{\eta_1\eta_2 \dots \eta_h}(x_{k_o}), \tag{7.7.20}$$

$D^{\ell}_{\eta_1\eta_2 \dots \eta_{i-1}\eta_i^*\eta_{i+1} \dots \eta_h}(x_{k_o}) =$

$$D^{\ell}_{\eta_1\eta_2 \dots \eta_h}(x_{k_o}), \tag{7.7.21}$$

(C) $$D^{\ell+1}_{\eta_1\eta_2 \cdots \eta_{i-1}\,\eta_i^*\,\eta_{i+1} \cdots \eta_h}(x_{k_o}) \subseteq D^{\ell+1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o}). \tag{7.7.22}$$

<u>Proof of (A)</u> Let r_i be an integer such that $\eta_i^{r_i}$ is the identity permutation. Then $\eta_i^{r_i-1}(x_t) = \eta_i^{r_i-1}(\eta_i(x_u)) = \eta_i^{r_i}(x_u) = x_u$ and $\eta_i^{r_i-1}(x_s) = \eta_i^{r_i-1}(\eta_i(x_u)) = \eta_i^{r_i}(x_v) = x_v$. Since $\eta_i^{r_i-1}$ is a symmetric permutation and $\vec{a}_s$ dominates $\vec{a}_t$, $\vec{a}_v$ dominates $\vec{a}_u$, by Lemma 7.6.3. By (7.7.16), $x_v \in D^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$. Then x_u is not a member of $H^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$, because, if x_u is a member, we will have x_u and x_v in $H^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ such that $\vec{a}_v$ dominates $\vec{a}_u$, contradicting that $\vec{a}_s$, does not dominate $\vec{a}_t$, for any pair of different variables x_s, and x_t, in $H^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$. Thus, (A) is proved.

<u>Proof of (B)</u> Since $x_u \notin H^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$,

$$\eta_{j^*_{k^*}} \circ \; j^*_{k-1} \circ \cdots \circ \eta_{j_1^*} \neq x_v \tag{7.7.24}$$

for any positive integer $k^* \leq \ell - 1$ and any j_1^* , j_2^* , ..., j_{k^*} in $\{1, 2, \ldots f\}$. For any j_p^* where $1 \leq k^* \leq \ell - 1$, if $\eta'_{j^*_p}$ is defined as

$$\eta'_{j^*_p} = \begin{cases} \eta'_{j^*_p} & \text{if } j_p^* \neq i, \\ \eta^*_{j^*_p} & \text{if } j_p^* = i. \end{cases} \tag{7.7.25}$$

By the definition of $\eta'_{j_1^*}$ in (7.7.25), by the definition of $\eta^*_{j_1^*}$ in (7.7.17), and by the fact that $x_{k_o} \neq x_u$ or x_v,

$$\eta'_{j^*_{k^*}} \circ \eta'_{j^*_{k^*-1}} \circ \ldots \circ \eta'_{j^*_2} \circ \eta'_{j^*_1} (x_{k_o})$$

$$= \eta'_{j^*_{k^*}} \circ \eta'_{j^*_{k^*-1}} \circ \ldots \circ \eta'_{j^*_2} \circ \eta_{j^*_1} (x_{k_o})$$

for any k^* such that $1 \leq k^* \leq \ell$ and any $j^*_1, j^*_2, \ldots, j^*_{k^*}$ in $\{1, 2, \ldots, h\}$. By the definition of $\eta'_{j^*_2}$ in (7.7.25), and by the definition of $\eta^*_{j^*_2}$ in (7.7.17),

$$\eta'_{j^*_{k^*}} \circ \ldots \circ \eta'_{j^*_3} \circ \eta'_{j^*_2} \circ \eta_{j^*_1} (x_{k_o})$$

$$= \eta'_{j^*_{k^*}} \circ \ldots \circ \eta'_{j^*_3} \circ \eta_{j^*_2} \circ \eta_{j^*_1} (x_{k_o}),$$

for any k^* such that $2 \leq k^* \leq \ell$ and any $j^*_1, j^*_2, \ldots, j^*_{k^*}$ in $\{1, 2, \ldots, h\}$, since, by (7.7.23) and (7.7.24), $\eta_{j^*_1}(x_{k_o}) \neq x_u$ or x_v. By the definition of $\eta'_{j^*_k}$ in (7.7.25), and by the definition of $\eta^*_{j^*_k}$ in (7.7.17),

$$\eta'_{j^*_k} \circ \eta_{j^*_{k-1}} \circ \ldots \circ \eta_{j^*_1} (x_{k_o})$$

$$= \eta_{j^*_k} \circ \eta_{j^*_{k-1}} \circ \ldots \circ \eta_{j^*_1} (x_{k_o})$$

for any k^* such that $1 \leq k^* \leq \ell$ and any $j^*_1, j^*_2, \ldots, j^*_{k^*}$ in $\{1, 2, \ldots, h\}$, since, by (7.7.23) and (7.7.24), $\eta_{j^*_{k^*-1}} \circ \ldots \circ \eta_{j^*_1} (x_{k_o}) \neq x_u$ or x_v for any k^* such that $1 \leq k^* \leq \ell$. Thus we have

$$\eta'_{j^*_{k^*}} \circ \eta'_{j^*_{k^*-1}} \circ \ldots \circ \eta'_{j^*_2} \circ \eta'_{j^*_1} (x_{k_o})$$

$$= \eta'_{j^*_{k^*}} \circ \eta'_{j^*_{k^*-1}} \circ \ldots \circ \eta'_{j^*_2} \circ \eta_{j^*_1} (x_{k_o})$$

$$= \eta'_{j^*_{k^*}} \circ \eta'_{j^*_{k^*-1}} \circ \ldots \circ \eta_{j^*_2} \circ \eta_{j^*_1} (x_{k_o})$$

.

.

.

$$= \eta'_{j^*_{k^*}} \circ \eta_{j^*_{k^*-1}} \circ \ldots \circ \eta_{j^*_2} \circ \eta_{j^*_1} (x_{k_o})$$

$$= \eta_{j^*_{k^*}} \circ \eta_{j^*_{k^*-1}} \circ \ldots \circ \eta_{j^*_2} \circ \eta_{j^*_1} (x_{k_o})$$

for any k^* such that $1 \leq k^* \leq \ell$ and any $j^*_1, j^*_2, \ldots, j^*_{k^*}$ in $\{1, 2, \ldots, h\}$. Therefore,

$$H^{\ell}_{\eta_1\eta_2\cdots\eta_{i-1}\eta^*_i\eta_{i+1}\cdots\eta_h}(x_{k_o}) = H^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}),$$

and $D^{\ell}_{\eta_1\eta_2\cdots\eta_{i-1}\eta_i\eta_{i+1}\cdots\eta_h}(x_{k_o}) = D^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$

are proved, i.e., (B) is proved.

<u>Proof of (C)</u> Let us show that each variable in $D^{\ell+1}_{\eta_1\eta_2\cdots\eta_{i-1}\eta^*_i\eta_{i+1}\cdots\eta_h}(x_{k_o})$ is also a variable in $D^{\ell+1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$. Since $D^{\ell}_{\eta_1\eta_2\cdots\eta_{i-1}\eta_i\eta_{i+1}\cdots\eta_h}(x_{k_o})$

$= D^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ from (7.7.21), and every variable in

$D^{\ell+1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ is mapped from some variable in $D^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$

by some symmetric permutation η_e where $1 \leq e \leq h$ by Theorem 7.7.1, we only have to show that if $\eta^*_i(x_{d'})$ is in $D^{\ell+1}_{\eta_1\eta_2\cdots\eta_{i-1}\eta^*_i\eta_{i+1}\cdots\eta_h}(x_{k_o})$

for some $x_{d'}$ in $D^{\ell}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$ then $\eta^*_i(x_{d'})$ is also in

$D^{\ell+1}_{\eta_1\eta_2 \cdots \eta_h}(x_{k_o})$.

Since x_u is not a member of $H^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ (from (A)), x_u is not a member of $D^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$, by (7.7.7). Since $x_{d'}$ is in $D^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$, $x_{d'} \neq x_u$. If $x_{d'} \neq x_v$, then $\eta^*_i(x_{d'}) = \eta_i(x_{d'})$ is in $D^{\ell+1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$, by (7.7.17). If $x_{d'} = x_v$ then $\eta^*_i(x_{d'})$ $= \eta^*_i(x_v) = x_t$, by (7.7.17). In the following, we shall show that x_t is a variable in $D^{\ell+1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$.

Since $\eta^*_i(x_{d'}) = x_t$ is in $D^{\ell+1}_{\eta_1\eta_2\cdots\eta_{i-1}\eta^*_i\eta_{i+1}\cdots\eta_h}(x_{k_o})$ and

$D^{\ell+1}_{\eta_1\eta_2\cdots\eta_{i-1}\eta^*_i\eta_{i+1}\cdots\eta_h}(x_{k_o}) = H^{\ell+1}_{\eta_1\eta_2\cdots\eta_{i-1}\eta^*_i\eta_{i+1}\cdots\eta_h}(x_{k_o})$

$- H^{\ell}_{\eta_1\eta_2\cdots\eta_{i-1}\eta^*_i\eta_{i+1}\cdots\eta_h}(x_{k_o})$, x_t is not in

$H^{\ell}_{\eta_1\eta_2\cdots\eta_{i-1}\eta^*_i\eta_{i+1}\cdots\eta_h}(x_{k_o})$. From (B), x_t is also not in

$H^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$. Since x_t is in $H^{\ell+1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ and x_t is not in

$H^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$, x_t must be in $D^{\ell+1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$, by (7.7.7). Thus (C) is proved.

From (B), $H^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ and $D^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ are not changed after η_i is modified to η_i^*. From (C), $D^{\ell+1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ will contain no more new variable after η_i is modified to η_i^*. If all η_i such that $\eta_i(x_u) = x_s$ for some x_v in $D^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ are modified to η^* as defined in (7.7.17) and denote the modified $\eta_1,\eta_2,\cdots,\eta_h$ as $\eta_1^*,\eta_2^*,\ \cdots,\ \eta_h^*$, then from (A) and (7.7.17),

$$x_s \notin H^{\ell+1}_{\eta_1^*\eta_2^*\cdots\eta_h^*}(x_{k_o}), \qquad (7.7.26)$$

proving (d).

From (7.7.20), (7.7.21), and the definition of η_1^*, η_2^*, $\cdots$, η_h^*,

$$H^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}) = H^{\ell}_{\eta_1^*\eta_2^*\cdots\eta_h^*}(x_{k_o}),$$

and

$$D^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}) = D^{\ell}_{\eta_1^*\eta_2^*\cdots\eta_h^*}(x_{k_o}),$$

proving (b) and (c) respectively.

From (7.7.22), (7.7.26) and the definition of $\eta_1^*,\eta_2^*,\cdots,\eta_h^*$,

$$D^{\ell+1}_{\eta_1^*\eta_2^*\ldots\eta_h^*} \subsetneqq D^{\ell+1}_{\eta_1^*\eta_2^*\cdots\eta_h^*}(x_{k_o}).$$

Q.E.D.

Lemma 7.7.7 Suppose there is no dominating row in the constraint matrix A of (P) and $\eta_1, \eta_2, \ldots, \eta_h$ are symmetric permutations of (P) such that column $\vec{a}_{s'}$ does not dominate column $\vec{a}_{t'}$ for every pair of variables $x_{s'}$ and $x_{t'}$ in $H^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$. If there esists some pair of variables x_s and x_t in $H^{\ell+1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ such that $\vec{a}_s$ dominates $\vec{a}_t$, then there exist symmetric permutations $\eta_1^+, \eta_2^+, \ldots, \eta_h^+$ such that

(1) $\eta_1^+, \eta_2^+, \ldots, \eta_h^+$ are obtained from $\eta_1, \eta_2, \ldots, \eta_h$ by repeating the modification described in (a) of Lemma 7.7.6 until no pair of variables x_{s^*} and x_{t^*} such that $\vec{a}_{s^*}$ dominates $\vec{a}_{t^*}$ exist in $H^{\ell+1}_{\eta_1^*\eta_2^*\cdots\eta_h^*}(x_{k_o})$,

(2) $\vec{a}_{s'}$ does not dominate $\vec{a}_{t'}$ for any pair of variables $x_{s'}$ and $x_{t'}$ in $H^{\ell+1}_{\eta_1^+\eta_2^+\cdots\eta_h^+}(x_{k_o})$,

(3) $H^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}) = H^{\ell}_{\eta_1^+\eta_2^+\cdots\eta_h^+}(x_{k_o})$ and $D^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}) = D^{\ell}_{\eta_1^+\eta_2^+\cdots\eta_h^+}(x_{k_o})$,

(4) $D^{\ell+1}_{\eta_1^+\eta_2^+\cdots\eta_h^+}(x_{k_o}) \subsetneq D^{\ell+1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$.

Proof Modifying $\eta_1, \eta_2, \ldots, \eta_h$ according to the modification described in (a) of Lemma 7.7.6, one obtains symmetric permutations $\eta_1^*, \eta_2^*, \ldots, \eta_h^*$ of (P) such that x_s is not a variable in $H^{\ell+1}_{\eta_1^*\eta_2^*\cdots\eta_h^*}(x_{k_o})$, by Lemma 7.7.6. Then $\eta_1^*, \eta_2^*, \ldots, \eta_h^*$ are symmetric

permutations satisfying properties (1) (3) and (4) of this lemma, by Lemma 7.7.6. If there exists no pair of variables x_{s*} and x_{t*} in $H^{\ell+1}_{\eta_1^*\eta_2^* \cdots \eta_h^*}(x_{k_o})$ such that $\vec{a}_{s*}$ dominates $\vec{a}_{t*}$, they satisfy property (2) also. Regarding these $\eta_1^*, \eta_2^*, \ldots, \eta_h^*$ as $\eta_1^+, \eta_2^+, \ldots, \eta_h^+$, this lemma is proved. If there exists a pair of variables x_{s_1} and x_{t_1} in $H^{\ell+1}_{\eta_1^*\eta_2^* \cdots \eta_h^*}(x_{k_o})$ such that $\vec{a}_{s_1}$ dominates $\vec{a}_{t_1}$, then one can repeat the same modification in (a) of Lemma 7.7.6, regarding $\eta_1^*, \eta_2^*, \ldots, \eta_h^*$ as $\eta_1, \eta_2, \ldots, \eta_h$. Each time the modification is made, at least one variable is deleted from $D^{\ell+1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$, by Lemma 7.7.6. Since the number of variables in $D^{\ell+1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ is finite, this process will lead to symmetric permutations $\eta_1^+, \eta_2^+, \ldots, \eta_h^+$ such that $\vec{a}_{s'}$ does not dominate $\vec{a}_{t'}$ for any pair of variables $x_{s'}$ and $x_{t'}$ in $H^{\ell+1}_{\eta_1^+\eta_2^+ \cdots \eta_h^+}(x_{k_o})$. Since $H^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ and $D^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ are not changed at each modification,

$$H^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}) = H^{\ell}_{\eta_1^+\eta_2^+\cdots\eta_h^+}(x_{k_o}),$$

and
$$D^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}) = D^{\ell}_{\eta_1^+\eta_2^+\cdots\eta_h^+}(x_{k_o}).$$

Since $D^{\ell+1}_{\eta_1^*\eta_2^*\cdots\eta_h^*}(x_{k_o}) \subsetneqq D^{\ell+1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ for each modification,

$$D^{\ell+1}_{\eta_1^+\eta_2^+\cdots\eta_h^+}(x_{k_o}) \subsetneq D^{\ell+1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}).$$

Q.E.D.

It will be proved in Lemma 7.7.8 that, the following procedure can be used to update symmetric permutations $\eta_1, \eta_2, \ldots, \eta_h$ of (P) such that the resulted permutations $\hat{\eta}_1, \hat{\eta}_2, \ldots, \hat{\eta}_h$ are still symmetric permutations of (P).

<u>Procedure $\widehat{GF}$</u> (x_{k_o}):

<u>$\hat{F}1$</u> $H \leftarrow$ empty, $D \leftarrow \{x_{k_o}\}$, $D1 \leftarrow$ empty

$\hat{F}2$ For each $i = 1, 2, \ldots, h$ and each x_v in D, do the following:

<u>$\hat{F}2.1$</u> $x_s \leftarrow \eta_i(x_v)$.

<u>$\hat{F}2.2$</u> If x_s is not a variable in H or in D and $\vec{a}_s$ does not dominate $\vec{a}_t$ for any x_t in H or in D, then store x_s in D1.

<u>$\hat{F}2.3$</u> If x_s is not a variable in H or in D and $\vec{a}_s$ dominates $\vec{a}_t$ for some x_t in H or in D then update η_i to η_i^* defined as

$$\eta_i^* : \begin{cases} x_d \longrightarrow \eta_i(x_d), \text{ if } x_d \neq x_u, x_v, \\ x_u \longrightarrow x_v, \\ x_v \longrightarrow x_t, \end{cases}$$

where x_u is the variable such that $\eta_i(x_u) = x_t$.

$\hat{F}3$ $H \leftarrow$ (union of H and D), $D \leftarrow D1$, $D1 \leftarrow$ empty.

$\hat{F}4$ If D is empty, then procedure terminates. Otherwise go to step G2. ∎

Lemma 7.7.8 Suppose there is no dominating row in the constraint matrix A of (P) and $\eta_1, \eta_2, \cdots, \eta_h$ are symmetric permutations of (P). Then

(1) for any given variable x_{k_o}, the last updated permutations $\eta_1, \eta_2, \cdots, \eta_h$, denoted by $\hat{\eta}_1, \hat{\eta}_2, \cdots, \hat{\eta}_h$, by the above $\widehat{GF}$($G_{\hat{\eta}_1\hat{\eta}_2\cdots\hat{\eta}_h}$ Finding) procedure are symmetric permutations of (P).

(2) column $\vec{a}_{s'}$ does not dominate column $\vec{a}_{t'}$ for any pair of variables $x_{s'}$ and $x_{t'}$ in $G_{\hat{\eta}_1\hat{\eta}_2\ldots\hat{\eta}_h}(x_{k_o})$.

Proof From Lemma 7.7.6, the permutation η^* of step $\hat{F}2.3$ is still a symmetric permutation of (P). So $\hat{\eta}_1, \hat{\eta}_2, \cdots, \hat{\eta}_h$ are symmetric permutations of (P). In the following we shall show that the set H in the procedure $\widehat{FG}$ is the set $G_{\hat{\eta}_1\hat{\eta}_2\cdots\hat{\eta}_h}(x_{k_o})$ after the procedure $\widehat{FG}$ is applied to x_{k_o}. Then, since column $\vec{a}_{s'}$ does not dominate $\vec{a}_{t'}$ for every pair of variables $x_{s'}$ and $x_{t'}$ in H (from the way it is constructed) the lemma is proved.

Suppose $H = H^{\ell-1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ and $D = D^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ before we go into step $\hat{F}2$ of the $\widehat{FG}$ procedure. Two cases may occur in step $\hat{F}2$.

Case (1) None of permutations $\eta_1, \eta_2, \cdots, \eta_h$ is updated in step $\hat{F}2.3$.

In this case, the procedure $\widehat{FG}$ goes through $\hat{F}2.1$ and $\hat{F}2.2$ only in $\hat{F}2$. After the procedure $\widehat{FG}$ goes through step $\hat{F}2$, the set $D^{\ell+1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ is the set D1, by Theorem 7.7.1. So after the procedure $\widehat{FG}$ goes through step $\hat{F}3$,

$$H = H^{\ell-1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}) + D^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}) = H^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}),$$

and $D = D^{\ell+1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$.

<u>Case (2)</u> Some of the permutations $\eta_1, \eta_2, \ldots, \eta_h$ are updated in step $\hat{F}2.3$.

In this case, the procedure $\widehat{FG}$ goes through $\hat{F}2.1$ and $\hat{F}2.3$ only in $\hat{F}2$. The updated symmetric permutations $\eta_1^+, \eta_2^+, \ldots, \eta_h^+$ have the following properties:

$$H^{\ell}_{\eta_1^+\eta_2^+\cdots\eta_h^+}(x_{k_o}) = H^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}), \qquad (7.7.27)$$

and $$D^{\ell}_{\eta_1^+\eta_2^+\cdots\eta_h^+}(x_{k_o}) = D^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}), \qquad (7.7.28)$$

by Lemma 7.7.7. From (7.7.28) and Theorem 7.7.1, $D^{\ell+1}_{\eta_1^+\eta_2^+\cdots\eta_h^+}(x_{k_o})$ is D1. So, after the procedure $\widehat{FG}$ goes through step $\hat{F}3$,

$$H = H^{\ell-1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}) + D^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}) = H^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$$

$$= H^{\ell}_{\eta_1^+\eta_2^+\cdots\eta_h^+}(x_{k_o}),$$

and $$D = D^{\ell+1}_{\eta_1^+\eta_2^+\cdots\eta_h^+}(x_{k_o}).$$

From cases (1) and (2), if $H = H^{\ell-1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ and $D = D^{\ell}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$, then after the procedure $\widehat{FG}$ goes through steps $\hat{F}2$ and $\hat{F}3$, $H = H^{\ell}_{\rho_1\rho_2\cdots\rho_h}(x_{k_o})$ and $D = D^{\ell+1}_{\rho_1\rho_2\cdots\rho_h}(x_{k_o})$, where $\rho_1, \rho_2, \ldots, \rho_h$ are $\eta_1, \eta_2, \ldots, \eta_h$ or $\eta_1^+, \eta_2^+, \ldots, \eta_h^+$.

In step F1 of the procedure $\widehat{FG}$, H is initialized as the empty set and D as $\{x_{k_o}\}$. After the procedure goes through steps $\hat{F}2$ and $\hat{F}3$ for the first time, $H = \{x_{k_o}\} = H^{o}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ and $D = D^{1}_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$. If D is empty, then

$$G_{\hat{\eta}_1\hat{\eta}_2\cdots\hat{\eta}_h}(x_{k_o}) = G_{\eta_1\eta_2\cdots\eta_h}(x_{k_o}) = \{x_{k_o}\} = H.$$

If $D \neq$ empty, by repeating steps $\hat{F}2$ and $\hat{F}3$, the procedure will arrive at symmetric permutations $\hat{\eta}_1, \hat{\eta}_2, \ldots, \hat{\eta}_h$, where $D = D^{\ell+1}_{\hat{\eta}_1\hat{\eta}_2\cdots\hat{\eta}_h}(x_{k_o})$ is empty for some k, such that $G_{\hat{\eta}_1\hat{\eta}_2\cdots\hat{\eta}_h}(x_{k_o}) = H^{k}_{\hat{\eta}_1\hat{\eta}_2\cdots\hat{\eta}_h}(x_{k_o}) = H$, since the number of variables in $X = \{x_1, x_2, \ldots, x_n\}$ is finite and the arguments in cases (1) and (2) can be repeatedly applied to sets H and D such that $H = H^{\ell-1}_{\rho_1\rho_2\cdots\rho_h}(x_{k_o})$ and $D = D^{\ell}_{\rho_1\rho_2\cdots\rho_h}(x_{k_o})$ for some positive integer ℓ and some symmetric permutations $\rho_1, \rho_2, \ldots, \rho_h$.

Q.E.D.

It will be proved in Theorem 7.7.9 that the following GDCDP (General Dominated Column Deletion Procedure) can be applied to reduce Problem (P).

Procedure GDCDP (General Dominated Column Deletion Procedure) :

GD1 Delete dominating rows from the constraint matrix A.

GD2 Find a dominated column $\vec{a}_{k_o}$. If no dominated column is found then procedure terminates.

GD3 Apply the $\widehat{GF}$ procedure to x_{k_o} and then update η_1, η_2, ..., η_h to $\hat{\eta}_1$, $\hat{\eta}_2$, ..., $\hat{\eta}_h$.

GD4 Delete all columns with their corresponding variables in $G_{\hat{\eta}_1\hat{\eta}_2\cdots\hat{\eta}_h}(x_{k_o})$ and set all variables in $G_{\hat{\eta}_1\hat{\eta}_2\cdots\hat{\eta}_h}(x_{k_o})$ to 0.

GD5 Update η_1, η_2, ..., η_h to η_1', η_2', ..., η_h', which are obtained by restricting η_1, η_2, ..., η_h to the problem reduced at step GD4 and then go to step GD2. ∎

Theorem 7.7.9 Let η_1, η_2, ..., η_h be symmetric permutations of Problem (P). Then the above GDCDP can be applied to reduce Problem (P). If (P9) denotes the reduced problem by applying the GDCDP procedure to (P), then the last updated permutations η_1, η_2, ..., η_h are symmetric permutations of (P9).

Proof Symmetric permutations η_1, η_2, ..., η_h are preserved during step GD1, by Lemma 7.6.1. In the step GD4, since $\vec{a}_{k_o}$ is dominated

by some other column, each of the columns with their corresponding variables in $G_{\hat{\eta}_1\hat{\eta}_2\cdots\hat{\eta}_h}$ is also dominated by some other column, by 7.7.5. (Note that some of these columns may dominate each other.) But because, by Lemma 7.7.8, column $\vec{a}_{s'}$ does not dominate column $\vec{a}_{t'}$ for any pair of variables $x_{s'}$ and $x_{t'}$ in $G_{\hat{\eta}_1\hat{\eta}_2\cdots\hat{\eta}_h}(x_{k_o})$, there is no subset of columns in $G_{\hat{\eta}_1\hat{\eta}_2\cdots\hat{\eta}_h}(x_{k_o})$ which dominate each other. Thus all columns with their corresponding variables in $G_{\hat{\eta}_1\hat{\eta}_2\cdots\hat{\eta}_h}(x_{k_o})$ can be deleted, without having wrong solutions by deleting columns which dominate each another. By Theorem 7.7.4, the permutations $\eta_1', \eta_2', \ldots, \eta_h'$ obtained by restricting $\eta_1, \eta_2, \ldots, \eta_h$ to the problem reduced at the step GD4 are symmetric permutations of this reduced problem.

Q.E.D.

From the above theorem, if the GDCDP procedure is applied to a given problem (P) with symmetric permutations $\eta_1, \eta_2, \ldots, \eta_h$, then the reduced Problem (P9) will have no dominating row and dominated column in its reduced constraint matrix. Also the reduced permutations $\eta_1', \eta_2', \ldots, \eta_h'$ obtained from $\eta_1, \eta_2, \ldots, \eta_h$ by the corresponding modification are symmetric permutations of (P9).

It will be proved in Theorem 7.7.10 that the following procedure GECFP (General Essential Column Finding Procedure) can be applied to reduce the Problem (P).

Procedure GECFP :

GE1 Find an essential column $\vec{a}_{k_o}$ of the constraint matrix. If no essential column is found, then the procedure terminates.

GE2 Apply the procedure FG to variable x_{k_o} and obtain $G_{\eta_1 \eta_2 \cdots \eta_h}(x_{k_o})$. Fix all variables in $G_{\eta_1 \eta_2 \cdots \eta_h}(x_{k_o})$ to 1 and delete all rows covered by the columns with their corresponding variables in $G_{\eta_1 \eta_2 \cdots \eta_h}(x_{k_o})$.

GE3 Update $\eta_1, \eta_2, \ldots, \eta_h$ to $\eta_1', \eta_2', \ldots, \eta_h'$ obtained by restricting $\eta_1, \eta_2, \ldots, \eta_h$ to this reduced problem (i.e., the problem obtained by fixing all variables in $G_{\eta_1 \eta_2 \cdots \eta_h}(x_{k_o})$ to 1) and go to step GE1. ∎

Theorem 7.7.10 Let (P) be a problem with symmetric permutations $\eta_1, \eta_2, \ldots, \eta_h$. Then the above GECFP can be applied to reduce the problem (P). If (P10) is the reduced problem obtained by applying the GECFP to the problem (P), then the last updated permutations $\eta_1, \eta_2, \ldots, \eta_h$ are symmetric permutations of (P10).

Proof $\eta_{j_k}^{p_k} \circ \eta_{j_{k-1}}^{p_{k-1}} \circ \cdots \circ \eta_{j_1}^{p_1}$ is a symmetric permutation of problem (P) for any positive integers k, $p_1, p_2, \ldots, p_k$ and any $\eta_{j_1}, \eta_{j_2}, \ldots, \eta_{j_k}$ in $\{\eta_1, \eta_2, \ldots, \eta_h\}$. Therefore, if $\vec{a}_{k_o}$ is an essential column, then all columns with their corresponding variables in

$G_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ are essential, by Lemma 7.6.8. Thus in solving (P) all columns with their corresponding variables in $G_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ can all be fixed to 1. By Theorem 7.7.4, the permutations η_1', η_2', ..., η_h' obtained by restricting η_1, η_2, ..., η_h to the reduced problem (problem obtained by fixing variables in $G_{\eta_1\eta_2\cdots\eta_h}(x_{k_o})$ to 1) are symmetric permutations of this reduced problem.

Q.E.D.

From Theorem 7.7.9 and 7.7.10, if the procedures GDCDP and GECFP are repeatedly applied to the problem (P) with symmetric permutations η_1, η_2, ..., η_h, then the reduced problem (P11), where none of the three reduction operations can be applied, is a symmetric problem with symmetric permutations η_1', η_2', ..., η_h', which are obtained from η_1, η_2, ..., η_h by the corresponding modification as described in Theorems 7.7.9 and 7.7.10.

7.8 Some Computational Results

The symmetric property of the minimal covering problem in the implicit enumeration algorithm discussed in this chapter are utilized in the algorithm of Section 5.3. A detail description of this is given in [32]. This section gives some computational comparison on solving problems with and without using the symmetric property of the given problem. Seven symmetric problems are tested. The constraint matrices of problems 1 and 2 are of the form (7.7.1),

where

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} ,$$

$$E = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} ,$$

$$F = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} ,$$

$$G = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} ,$$

for the problem 1 and

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix},$$

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$F = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$G = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

for the problem 2. Problem 3 is the testing problem IBM No. 9 in [15]. Its constraint matrix is

$$A = \left(\begin{array}{c|c|c} B & D & 0 \\ \hline C & D & 0 \\ \hline 0 & B & D \\ \hline 0 & C & D \\ \hline D & 0 & B \\ \hline D & 0 & C \\ \hline D & D & D \end{array}\right) ,$$

where

$$B = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} ,$$

$$C = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} ,$$

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

It can be proved by Theorem 7.4.1 that the permutation defined by

$$\eta \quad : \begin{cases} x_i \longrightarrow x_{i+10} & \text{if } 1 \leq i \leq 5 \\ x_i \longrightarrow x_{i-5} & \text{if } 6 \leq i \leq 15 \end{cases}$$

is a symmetric permutation of this problem. Problem 4 is the smaller* one of the two difficult problems reported in [24]. Its constraint matrix A is

$$A_{27} = \left[\begin{array}{c|c|c} A_9 & 0 & 0 \\ \hline 0 & A & 0 \\ \hline 0 & 0 & A_9 \\ \hline \tilde{C}^1 & I & \tilde{P}^1 \\ \hline \vdots & \vdots & \vdots \\ \hline \tilde{C}^9 & I & \tilde{P}^9 \end{array}\right],$$

---

* See the footnote in Section 9.2, page 162

where

$$A_9 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix},$$

I is the 9x9 identity matrix, $\tilde{C}^k$ is a 9x9 zero-one matrix with elements in the k-th column equal 1 and all other elements equal 0 for k = 1, 2, ..., 9, $\tilde{P}^k$ is a 9x9 permutation matrix for k = 1, 2, ..., 9, and 0 is the matrix with all 0 elements. A complete description of this problem is given in [24]. It can be shown by Theorem 7.4.1 that the permutation η defined by

$$\eta : \begin{cases} x_i \longrightarrow x_{i+3}, & \text{if } 1 \leq i \leq 6,\ 10 \leq i \leq 15,\ 19 \leq i \leq 24, \\ x_i \longrightarrow x_{i-3}, & \text{if } 7 \leq i \leq 9,\ 16 \leq i \leq 18,\ 25 \leq i \leq 27, \end{cases}$$

is a symmetric permutation of this problem. Problem 5 and 6 are problems obtained from minimizing the logic expression of seven-variable switching functions. Both switching functions are partially symmetric in variables y_1, y_2, and y_3. There are 3 symmetric permutations (derived from exchanging pairs of switching variables (y_1, y_2), (y_1, y_3) and (y_2, y_3)) for each of these 2 problems. Problem 7 is obtained from minimizing the logic expression of the totally symmetric six-variable switching function $S^6_{2,3,4}$ (a switching function whose value is 1 if exactly 2, 3, or 4 input variables are 1). There are 15 symmetric permutations (derived from exchanging each pair of variables) provided for this problem. The computational results of solving these seven problems, both with and without using the symmetric property of the given problem , are shown in the Table 7.8.1. The computer used for obtaining these results is IBM360/75J. The column under "No Of ITER.", "No Of BKTRK", "TIME IN SEC." are explained in Table 5.4.1. "?" in the table shows that the figure in that field is not known.

From this computational comparison, one can see that utilization of the symmetric property of the given problem yields better computational results. Computational improvement through the utilization of the symmetric property is more than ten times for problem 7.

PROB.	PROB. SIZE		WITHOUT USING SYMMETRIC			USING SYMMETRIC PROPERTY		
NO.	m	n	NO. OF ITER.	NO. OF BKTRK	TIME IN SEC.	NO. OF ITER.	NO. OF BKTRK	TIME IN SEC.
1	45	45	497	406	10.89	327	269	6.92
2	54	54	20987	12320	?*	6007	4676	122.72
3	35	15	148	87	1.07	86	50	0.57
4	117	27	6321	3063	94.14	4246	2348	58.47
5	112	87	327	198	12.29	143	90	5.33
6	113	69	301	188	11.87	187	125	8.37
7	50	90	>225000	?	> 5400**	19620	13401	453.23

Table 7.8.1

Comparison of two cases : with or without using the symmetric property of the given problem.

* It took about 62.87 seconds on the CDC Cyber/175 computer.

** It took about 921 seconds on CDC Cyber/175 to run that many iterations. The operation speed of CDC Cyber/175 computer is estimated more than six times faster than that of the IBM 360/75J computer.

8. PERMUTATIONAL PRECLUDING PROCEDURE

Let η be a general permutation (not necessarily symmetric) on $X = \{x_1, x_2, \ldots, x_n\}$. In solving the problem (P) by the implicit enumeration method, if the subproblem with x_i fixed to 1 has been enumerated and if $x_j = \eta(x_i)$, then in the subproblem with x_i fixed to 0 and with x_j fixed to 1, a better feasible solution (a feasible solution better than the best solution obtained so far) can only be found as $(x_1, x_2, \ldots, x_n)$ such that its corresponding $(\eta(x_1), \eta(x_2), \ldots, \eta(x_n))$ is not a feasible solution. In this chapter, properties of this kind of feasible solutions are persued. Then these properties are used to preclude some subproblems where no better feasible solution can be found. The discussion in Section 5.2 for precluding subproblems in the enumeration procedure becomes a special case of this chapter.

8.1 Generalized E-sets

Theorem 8.1.1 Let η be a permutation (not necessarily symmetric) on $\{x_1, x_2, \ldots, x_n\}$. If $(x_1, x_2, \ldots, x_n)$ is a feasible solution of the problem (P) such that $(\eta(x_1), \eta(x_2), \ldots, \eta(x_n))$ is not a feasible solution, then there exists some row $\vec{r}_k = (a_{k1}, a_{k2}, \ldots, a_{kn})$ of A such that

(1) $\vec{r}_k$ does not dominate any row of $\eta(A)$ (see Section 7.4 for the definition of $\eta(A)$),

(2) if $a_{k\ell} = 1$ in $\vec{r}_k$, then $\eta(x_\ell) = 0$.

Furthermore if $\eta(x_i) = x_j = 1$, then a_{ki} is 0.

<u>Proof</u> Since $(x_1, x_2, \ldots, x_n)$ is a feasible solution of (P),

$$A \cdot \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{pmatrix} . \qquad (8.1.1)$$

Rewrite the above inequality as

$$\eta(A) \cdot \begin{pmatrix} \eta(x_1) \\ \eta(x_2) \\ \cdot \\ \cdot \\ \cdot \\ \eta(x_n) \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{pmatrix} . \qquad (8.1.2)$$

Since $(\eta(x_1), \eta(x_2), \ldots, \eta(x_n))$ is not a feasible solution,

$$A \cdot \begin{pmatrix} \eta(x_1) \\ \eta(x_2) \\ \cdot \\ \cdot \\ \cdot \\ \eta(x_n) \end{pmatrix} \not\geq \begin{pmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{pmatrix} , \qquad (8.1.3)$$

i.e., there exists some row $\vec{r}_k$ of A such that

$$a_{k1} \cdot \eta(x_1) + a_{k2} \cdot \eta(x_2) + \ldots + a_{kn} \cdot \eta(x_n) = 0. \qquad (8.1.4)$$

The above equality shows

(1) $\vec{r}_k$ does not dominate any row of $\eta(A)$. (Otherwise from (8.1.2), $a_{k1}\cdot\eta(x_1) + a_{k2}\cdot\eta(x_2) + \ldots + a_{kn}\cdot\eta(x_n) \geq 1$ holds)

(2) if $a_{k\ell} = 1$, then $\eta(x_\ell) = 0$.

Furthermore if $\eta(x_i) = x_j = 1$, then a_{ki} must be 0, since $a_{ki} = 1$ implies $\eta(x_i) = 0$ by (2).

Q.E.D.

For a given permutation η and a given index j let $E_\eta(j)$ be the set of rows of A satisfying condition (1) of Theorem 8.1.1 and having their i-th components equal 0, where i is the index of the variable x_i such that $\eta(x_i) = x_j$. $E_\eta(j)$ is defined as the <u>generalized E-set</u> of column $\vec{a}_j$ with respect to the permutation η.

<u>Example 8.1.1</u> Let the constraint matrix A of the problem (P) be

$$A = \begin{bmatrix} 1&1&0&0&1&0 \\ 1&0&1&0&0&0 \\ 0&0&1&1&1&0 \\ 1&1&0&1&0&0 \\ 0&1&1&0&0&1 \\ 0&0&0&0&1&1 \end{bmatrix} . \qquad (8.1.5)$$

Define two permutations η_1 and η_2 on $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ as

$$\eta_1 : \begin{cases} x_1 \longrightarrow x_3, \\ x_2 \longrightarrow x_2, \\ x_3 \longrightarrow x_1, \\ x_4 \longrightarrow x_4, \\ x_5 \longrightarrow x_5, \\ x_6 \longrightarrow x_6, \end{cases} \qquad (8.1.6)$$

$$\eta_2 \; : \; \begin{cases} x_1 \longrightarrow x_3 \, , \\ x_2 \longrightarrow x_6 \, , \\ x_3 \longrightarrow x_1 \, , \\ x_4 \longrightarrow x_4 \, , \\ x_5 \longrightarrow x_2 \, , \\ x_6 \longrightarrow x_5 \, . \end{cases} \tag{8.1.7}$$

From the definition of $\eta_1(A)$ and $\eta_2(A)$,

$$\eta_1(A) = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{8.1.8}$$

$$\eta_2(A) = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} . \tag{8.1.9}$$

Then $E_{\eta_1}(3) = \{\vec{r}_3, \vec{r}_5\}$ and $E_{\eta_2}(3) = \{\vec{r}_5, \vec{r}_6\}$. ■

8.2 Precluding Of Subproblems

Precluding of subproblems using the properties stated in Theorem 8.1.1 is discussed in this section.

Let η^{-1} denote the inverse mapping of η and let $\eta^{-1}(a_i)$ denote the column a_j such that $\eta^{-1}(x_i) = x_j$.

Example 8.2.1 Let η_2 be the permutation defined in (8.1.7) and A be the constraint matrix (8.1.5), then

$$\eta^{-1} : \begin{cases} x_1 \longrightarrow x_3 , \\ x_2 \longrightarrow x_5 , \\ x_3 \longrightarrow x_1 , \\ x_4 \longrightarrow x_4 , \\ x_5 \longrightarrow x_6 , \\ x_6 \longrightarrow x_2 , \end{cases} \tag{8.2.1}$$

and $\eta_2^{-1}(\vec{a}_2) = \vec{a}_5$. ∎

<u>Theorem 8.2.1</u> Let $E_\eta(j)$ be the generalized E-set of $\vec{a}_j$ with respect to permutation η and S be a partial solution with x_j, x_{j_1}, x_{j_2}, ..., x_{j_r} fixed to 1. If each row in $E_\eta(j)$ is covered by some of $\eta^{-1}(\vec{a}_{j_1})$, $\eta^{-1}(\vec{a}_{j_2})$,..., $\eta^{-1}(\vec{a}_{j_r})$, then no row in $E_\eta(j)$ satisfies the condition (2) stated in Theorem 8.1.1 for any feasible completion of S.

Furthermore, if $x_j = \eta(x_i) = 1$, then $(\eta(x_1), \eta(x_2),..., \eta(x_n))$ is a feasible solution of (P) for every feasible completion $(x_1, x_2, ..., x_n)$ of S.

<u>Proof</u> Suppose $\vec{r}_k$ is a row in $E_\eta(j)$ satisfying condition (2) of Theorem 8.1.1 for some feasible completion $(x_1, x_2, ..., x_n)$ of S. Define $\eta^{-1}(j_t)$ to be the index of the variable x_{k_t} such that $\eta^{-1}(x_{j_t}) = x_{k_t}$ for t = 1, 2, ..., r. Since each row in $E_\eta(j)$ is covered by some of $\eta^{-1}(\vec{a}_{j_1})$, $\eta^{-1}(\vec{a}_{j_2})$, ..., $\eta^{-1}(\vec{a}_{j_r})$, row $\vec{r}_k$ is covered by some of $\eta^{-1}(\vec{a}_{j_1})$, $\eta^{-1}(\vec{a}_{j_2})$, ..., $\eta^{-1}(\vec{a}_{j_r})$. So at least one of $a_{k\eta^{-1}(j_1)}$, $a_{k\eta^{-1}(j_2)}$, ..., $a_{k\eta^{-1}(j_r)}$, say $a_{k\eta^{-1}(j_i)}$, must be 1. Since $a_{k\eta^{-1}(j_i)} = 1$, $\eta(x_{\eta^{-1}(j_i)}) = \eta(\eta^{-1}(x_{j_i})) = x_{j_i}$ must be 0, by the condition (2) of Theorem 8.1.1. This contradicts that x_{j_1}, x_{j_2},

$\ldots, x_{j_r}$ are fixed to 1 in S.

Furthermore, if $x_j = \eta(x_i) = 1$, and if there is a feasible completion $(x_1, x_2, \ldots, x_n)$ of S such that $(\eta(x_1), \eta(x_2), \ldots, \eta(x_n))$ is not a feasible solution, then, by Theorem 8.1.1, there exists some row r_k of A satisfies the conditions stated in that Theorem. Since $x_j = \eta(x_i) = 1$, $a_{ki} = 0$ by Theorem 8.1.1. By definition, row $\vec{r}_k$ must be a row in $E_\eta(j)$. Since there is no row in $E_\eta(j)$ satisfies condition (2) of Theorem 8.1.1, $\vec{r}_k$ can not be a row in $E_\eta(j)$. This is a contradiction.

Q.E.D.

Let η be a permutation on $\{x_1, x_2, \ldots, x_n\}$ such that $\eta(x_i) = x_j$. After the subproblem with x_i fixed to 1 has been enumerated, in the subproblem with x_i fixed to 0 and x_j fixed to 1, one can test whether each row in the generalized E-set, $E_\eta(j)$, of $\vec{a}_j$ with respect to η is covered by some of columns $\eta^{-1}(a_{j_1})$, $\eta^{-1}(a_{j_2})$, $\ldots$, $\eta^{-1}(\vec{a}_{j_r})$, where $j_1, j_2, \ldots, j_r$ are indices of the variables which are fixed to 1 and are not equal to x_j in the current partial solution S. If each row in $E_\eta(j)$ is covered by some of $\eta^{-1}(a_{j_1})$, $\eta^{-1}(a_{j_2})$, $\ldots$, $\eta^{-1}(a_{j_r})$, then $(\eta(x_1), \eta(x_2), \ldots, \eta(x_n))$ is also a feasible solution for every feasible completion $(x_1, x_2, \ldots, x_n)$ of the current partial solution S, by Theorem 8.2.1. Since the subproblem with x_i fixed to 1 has been enumerated, and $(\eta(x_1), \eta(x_2)$, $\eta(x_n))$ is a feasible solution with $\eta(x_i) = x_j = 1$, every feasible completion of S can not be better than the best solution obtained so

far. So the current subproblem can be skipped without losing a better feasible solution.

As a special case, let us consider the permutation defined by

$$\eta : \begin{cases} x_i \longrightarrow x_j , \\ x_j \longrightarrow x_i , \\ x_k \longrightarrow x_k , \quad \text{if } k \neq i, j \end{cases} \tag{8.2.2}$$

for some i and j. From the definition of η,

$$\left.\begin{aligned} \eta^{-1}(\vec{a}_i) &= \vec{a}_j , \\ \eta^{-1}(\vec{a}_j) &= \vec{a}_i , \\ \eta^{-1}(\vec{a}_{j_\ell}) &= \vec{a}_{j_\ell} , \quad \text{if } j_\ell \neq i, j \end{aligned}\right\} \tag{8.2.3}$$

<u>Theorem 8.2.2</u> The generalized E-set, $E_\eta(j)$, of the column $\vec{a}_j$ with respect to the particular η defined by (8.2.2) is a subset of E_{ij}, the E-set of column $\vec{a}_j$ with respect to column $\vec{a}_i$ (See Section 5.2 for the definition of E-set).

<u>Proof</u> From the definition of $E_\eta(j)$, each row of $E_\eta(j)$ does not dominate any row of $\eta(A)$ and each row of $E_\eta(j)$ must have its i-th element equal to 0, where i is the index of variable x_i such that $\eta(x_i) = x_j$. From the definition of η, (8.2.2), the only rows of A that may not dominate any row of $\eta(A)$ are those with their i-th and j-th elements different (i.e., one is 0 and the other is 1). Since each row of $E_\eta(j)$ must have its i-th element equal to 0, the only

rows that may be in $E_\eta(j)$ are those with their i-th element equal to 0 and the j-th element equal to 1, i.e., those rows in E_{ij}.

Q.E.D.

Example 8.2.1 Let us consider a problem with the constraint matrix

$$A = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} .$$

From the definition of $E_{1\,2}$, $E_{1\,2} = \{r_5, r_6\}$. Define a permutation η on $\{x_1, x_2, x_3, x_4, x_5\}$ as

$$\eta : \begin{cases} x_1 \longrightarrow x_2 , \\ x_2 \longrightarrow x_1 , \\ x_3 \longrightarrow x_3 , \\ x_4 \longrightarrow x_4 , \\ x_5 \longrightarrow x_5 , \end{cases}$$

then

$$\eta(A) = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} .$$

Since $\vec{r}_6$ of A dominates $\vec{r}_1$ of $\eta(A)$, $E_\eta(2) = \{ r_5 \}$. ∎

From (8.2.3) and Theorem 8.2.1, we obtain the following Corollary.

<u>Corollary 8.2.4</u> Let η be a permutation defined on $\{ x_1, x_2, \ldots, x_n \}$ as

$$\eta : \begin{cases} x_i \longrightarrow x_j , \\ x_j \longrightarrow x_i , \\ x_\ell \longrightarrow x_\ell , \quad \text{if } \ell \neq i, j \end{cases} \qquad (8.2.4)$$

If S is a partial solution with $x_j, x_{j_1}, x_{j_2}, \ldots, x_{j_r}$ fixed to 1, and if each row in $E_\eta(j)$ is covered by some of columns $\vec{a}_{j_1}, \vec{a}_{j_2}, \ldots, \vec{a}_{j_r}$, then $(\eta(x_1), \eta(x_2), \ldots, \eta(x_n))$ is a feasible solution of (P) for every feasible completion $(x_1, x_2, \ldots, x_n)$ of S.

<u>Proof</u> From (8.2.3), $\eta^{-1}(\vec{a}_{j_1}), \eta^{-1}(\vec{a}_{j_2}), \ldots, \eta^{-1}(\vec{a}_{j_r})$ in Theorem 8.2.1 can be replaced by $\vec{a}_{j_1}, \vec{a}_{j_2}, \ldots, \vec{a}_{j_r}$. This corollary is proved by replacing $\vec{a}_{j_1}, \vec{a}_{j_2}, \ldots, \vec{a}_{j_r}$ for $\eta^{-1}(\vec{a}_{j_1}), \eta^{-1}(\vec{a}_{j_2}), \ldots, \eta^{-1}(\vec{a}_{j_r})$ in Theorem 8.2.1.

Q.E.D.

Since $E_\eta(j)$ is a subset of E_{ij} for the permutation η of (8.2.4), the two conditions of Corollary 8.2.3 are more easily satisfied than the two conditions for E_{ij} given in Theorem 5.2.1. Theoretically, the computational efficiency of the algorithm described in Section 5.3 can further be improved if the conditions given in Corollary 8.2.3 are checked instead of the conditions given in

Theorem 5.2.1 for each partial solution S (i.e., E_{ij} is replaced by $E_\eta(j)$ in the testing set generated in step M4.4). But in actual programming experiment, it was found that most of the time the generalized E-set, $E_\eta(j)$, with respect to the particular permutation η of (8.2.4) for a subproblem with $x_i = 0$ and $x_j = 1$ is not different from the E-set E_{ij} for that subproblem. Consequently no example of actual computational improvement was found through the implementation of this generalized E-set with respect to this particular permutation.

No experiment concerning the checking of generalized E-sets with respect to general permutations has been done. This is would need further research.

9. THE MINIMAL COVERING PROBLEM WITH PARTITONED CONSTRAINT MATRIX

In this chapter, we consider solving the minimal covering problem (P) with a constraint matrix of the following form:

$$A = \left[\begin{array}{c|c|c|c} A_1 & & & \\ \hline & A_2 & & \\ \hline & & \ddots & \\ \hline & & & A_r \\ \hline C_1 & C_2 & \cdots & C_r \end{array}\right], \tag{9.1}$$

where A_i is a m_i by n_i zero-one matrix for $i = 1, 2, \ldots, r$, C_i is a c by n_i zero-one matrix for $i = 1, 2, \ldots, r$, and all other parts of A are all zero elements. The two problems reported in [24] are problem of this type.

A structure of the following form

$$A = \left[\begin{array}{c|c|c|c|c} A_1 & & & & \\ \hline & A_2 & & & \\ \hline & & \ddots & & \\ \hline & & & A_{r-1} & \\ \hline C_1 & C_2 & \cdots & C_{r-1} & C_r \end{array}\right] \tag{9.2}$$

is considered as a special case of the structure (9.1), in which A_r is a matrix with $m_r = 0$.

In solving the logic minimization problem for a multiple-output switching function, if the rows (corresponding to the true vectors) of the prime implicant table are rearranged such that true vectors implying more output functions are placed at the bottom of the table, then the constraint matrix of the formulated minimal covering problem will be of the form (9.2).

Example 9.1 Let us consider the problem of Example 2.2.2 again. If the true vectors $\vec{y}_3, \vec{y}_4, \vec{y}_5, \vec{y}_7, \vec{y}_{11}, \vec{y}_{12}, \vec{y}_{14}, \vec{y}_{16}$ are moved to the bottom of the prime implicant table, then this table becomes as follows:

$$
\begin{array}{c}
\\
\vec{y}_1 \\ \vec{y}_2 \\ \vec{y}_6 \\ \vec{y}_8 \\ \vec{y}_9 \\ \vec{y}_{10} \\ \vec{y}_{13} \\ \vec{y}_{15} \\ \vec{y}_3 \\ \vec{y}_4 \\ \vec{y}_5 \\ \vec{y}_7 \\ \vec{y}_{11} \\ \vec{y}_{12} \\ \vec{y}_{14} \\ \vec{y}_{16}
\end{array}
\begin{array}{c}
\begin{array}{cccccccccccc}
q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 & q_9 & q_{10} & q_{11} & q_{12}
\end{array} \\
\left[\begin{array}{cccc:cccc:cccc}
0 & 0 & 1 & 0 & & & & & & & & \\
0 & 0 & 1 & 1 & & & 0 & & & 0 & & \\
0 & 1 & 0 & 0 & & & & & & & & \\
1 & 1 & 0 & 0 & & & & & & & & \\
\hdashline
 & & & & 0 & 0 & 1 & 0 & & & & \\
 & & 0 & & 0 & 0 & 1 & 0 & & 0 & & \\
 & & & & 1 & 0 & 0 & 0 & & & & \\
 & & & & 1 & 0 & 0 & 0 & & & & \\
\hdashline
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1
\end{array}\right]
\end{array}
\quad .
\qquad \blacksquare
$$

The utilization of this special structure of constraint matrix in solving problem is discussed in this chapter. Some computational efficiency improvement through this utilization is shown by examples.

9.1 Upper Bounds On The Values Of Groups Of Variables

A minimal covering problem of this type can be restated as follows:

minimize $x_1+x_2+\ldots+x_{n_1}+x_{n_1+1}+\ldots+x_{n_1+n_2}+\ldots+x_n$,

subject to

$$A_1.\vec{x}_1 \geq \begin{pmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{pmatrix}_{m_1}$$

$$A_2.\vec{x}_2 \geq \begin{pmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{pmatrix}_{m_2}$$

$$\vdots$$

$$A_r.\vec{x}_r \geq \begin{pmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{pmatrix}_{m_r}$$

$$C_1 \cdot x_1 + C_2 \cdot x_2 + \ldots + C_r \cdot x_r \geq \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}_c .$$

$$x_i = 0 \text{ or } 1 \text{ for } i = 1, 2, \ldots, n,$$

where

$$x_1 = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_{n_1} \end{bmatrix}, \quad x_2 = \begin{bmatrix} x_{n_1+1} \\ x_{n_2+2} \\ \cdot \\ \cdot \\ \cdot \\ x_{n_1+n_2} \end{bmatrix}, \ldots \quad x_r = \begin{bmatrix} x_{n_1+n_2+\ldots+n_{r-1}+1} \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}.$$

To utilize the special structure of the above problem, variables $x_1, x_2, \ldots, x_n$ are first grouped into r groups as

$$G_1 = \{ x_1, x_2, \ldots, x_{n_1} \},$$

$$G_2 = \{ x_{n_1+1}, \ldots, x_{n_1+n_2} \},$$

$$\vdots$$

$$G_r = \{ x_{n_1+n_2+\ldots+n_{r-1}+1}, \ldots, x_n \}.$$

then an upper bound of the value of each group will be found and these upper bounds are used to preclude some unnecessary search in enumerating the problem.

Let us first see some definitions and a theorem which will be used later.

For k = 1, 2, ..., r, let P_k denote the following problem:

minimize $x_1+x_2+\ldots+x_n$,

subject to

$$(P_k) \qquad A_k \cdot X_k \geq \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}_{m_k} ,$$

$$x_i = 0 \text{ or } 1 \text{ for } i = 1, 2, \ldots, n,$$

and Z_k be its optimal value.

Theorem 9.1 For a given upper bound ZBAR on the optimal value of the problem (P), define

$$u_i = \text{ZBAR} - \sum_{\substack{j=1 \\ j \neq i}} z_j , \qquad (9.3)$$

for $i = 1, 2, \ldots, r$. If S is a partial solution with u_i variables in group G_i fixed to 1 for some i, then the value for any feasible completion of S is greater than or equal to ZBAR.

Proof None of the constraints in

$$A_k \cdot x_k \geq \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}_{m_k} , \text{ for } k = 1, 2, \ldots, i-1, i+1, \ldots, r,$$

is satisfied by only fixing variables in G_i to 1. In order to satisfy these constraints, at least $z_1 + z_2 + \ldots + z_{i-1} + z_{i+1} + \ldots + z_r$ variables not in G_i must be fixed to 1. If U_i variables of G_i are

already fixed to 1 in S, then any feasible completion of S must have value greater than or equal to

$$u_i+z_1+z_2+\ldots+z_{i-1}+z_{i+1}+\ldots+z_r,$$

which is equal to ZBAR, by (9.3).

Q.E.D.

Now the utilization of the special structure of the problem is described as follows.

In enumerating the problem, if there exists some i such that u_i-1 variables of G_i are fixed to 1 in the current partial solution, then all free variables in G_i must be fixed to 0, by Theorem 9.1, in order to get a feasible solution with objective value smaller than ZBAR, an upper bound of the optimal value of the problem. From this, one can see that U_i is an upper bound on the value of the group G_i for each i.

The current subproblem may become infeasible when free variables in G_i are fixed to 0. In this case, the current subproblem cannot have any feasible completion with a value smaller than ZBAR and program backtracks.

If there exists some i such that more than $U_i - 1$ variables of G_i are fixed to 1 in the current partial solution, then the program may backtrack immediately, since no feasible completion with objective value smaller than ZBAR can be found under the current partial solution, by Theorem 9.1.

When an improved upper bound ZBAR on the optimal value of the problem is found in the enumeration procedure, u_i for each group

G_i is updated by (9.3) for i = 1, 2, ..., r.

The above diccussion of the utilization of the special structure of the problem can easily be incorporated into the algorithm of Section 5.3. A detail description of this incorporation is given in [32].

9.2 Some Computational Results

Three problems of the type discussed in Section 9.1 are tested by this incorporated algorithm. Computational results are shown in Table 9.2.1, along with the results for the same three problems obtained by using algorithm in Section 5.3. Programs are coded in FORTRAN and compiled by FORTRAN H compiler. Problems are tested on IBM360/75J computer. All the columns in this table are explained in Table 5.4.1.

PROB. NO.	CHECKING UPPER BOUND FOR EACH GROUP VARIABLES			WITHOUT CHECKING UPPER BOUND FOR EACH GROUP VARIABLES		
	NO. OF ITER.	NO. OF BKTRK	TIME IN SEC	NO. OF ITER	NO. OF BKTRK	TIME IN SEC
1	5647	2963	77.25	6321	3063	94.14
2	4195	2738	161.83	4818	3204	200.89
3	8765	5008	246.39	12425	6554	356.37

Table 9.2.1

Comparison of two cases: with and without checking upper bound for each group of variables.

The first problem tested is the smaller one, A27, of the two difficult problems reported in [24]*. The constraint matrix of this problem is as follows:

$$A = \left[\begin{array}{c|c|c} A_1 & 0 & 0 \\ \hline 0 & A_2 & 0 \\ \hline 0 & 0 & A_3 \\ \hline C_1 & C_2 & C_3 \end{array}\right] ,$$

where A_1, A, A_3 are the same matrices of size 12x9 each, and $[C_1, C_2, C_3]$ is a 81x27 matrix, z_1, z_2, z_3 for the three smaller problems P_1, P_2, P_3 are 5.

The constraint matrix of the second problem is as follows:

$$\begin{array}{cc} & \begin{array}{cc} f_1 \qquad f_2 & f_1 \cdot f_2 \end{array} \\ A = & \left[\begin{array}{c|c} A_1 & 0 \\ \hline C_1 & C_2 \end{array}\right] \end{array} .$$

It is obtained from a prime implicant table of a six-variable switching function with two outputs f_1 and f_2 by permuting its rows. Here C_2 is the prime implicant table of the switching function $f_1 \cdot f_2$.

---

* The optimal value of the larger one, A45, of the two problems in [24] is proved by this program to be 30 in about 135 minutes, with 227,676 iterations. If the symmetric property (see Chapter 7), of this problem is taken into consideration, it can be proved in about 90 minutes with about 159,500 iterations.

It is a 84x36 matrix. $\begin{bmatrix} A \\ \hdashline C_1 \end{bmatrix}$ is the concatination of the prime implicant tables of f_1 and f_2. It took only few centiseconds to get $Z_1 = 8$ for the problem P_1. In solving this problem, Z_2 must be set to 0.

The third problem tested is constructed by the author. its constraint matrix is

$$A = \left[\begin{array}{c:c} A_1 & 0 \\ \hdashline 0 & A_2 \\ \hdashline C_1 & C_2 \end{array}\right]$$

where A_1 is a 50x60 matrix, A_2 is a 39x55 matrix and $[C_1, C_2]$ is a 11x115 matrix. It took about 2.4 seconds to get z_1 =15 for the problem P_1 and 0.4 seconds to get $z_2 = 8$ for the problem P_2.

From the computational results shown in Table 9.2.1, about 30% of computation time can be saved in solving the minimal covering problem with partitioned constraint matrix if the number of variables fixed to 1 in each group is checked to see if it exceeds its upper bound in the enumeration procedure.

10. THE GENERAL COST MINIMAL COVERING PROBLEM

This chapter discusses a generalization of the algorithm described in the previous chapters for the general cost minimal covering problem.

The <u>general cost minimal covering problem</u> (GP) is defined as follows:

$$\text{minimize } c_1x_1+c_2x_2+\ldots+c_nx_n ,$$

subject to

$$\text{(GP)} \qquad A \cdot \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} ,$$

$$x_i = 0 \text{ or } 1, \text{ for } i = 1, 2,\ldots,n,$$

where $A = (a_{ij})$ with $a_{ij} = 0$ or 1, and c_i is a non-zero positive integer. c_i is called the <u>cost</u> of the variable x_i. The minimal covering problem (P) defined in Chapter 3 is a special case of this problem, where the cost of each variable is 1.

In implementing a switching function using PLA, the size of PLA is first minimized by minimizing the number of terms used in expressing this function. Then depending on different technologies

used for implementation [30], one may want to minimize or maximize the number of contacts required at the intersections of horizontal and vertical lines. (If contacts between horizontal and vertical lines are formed, MOSFETs or diodes at the intersections become responsive to their input voltages.) The minimization (or maximization) of the number of contacts improves reliability of PLA. Suppose the switching function $f = \{f_1, f_2, \ldots, f_u\}$ to be implemented is expressed in the following disjunctive forms:

$$
\begin{aligned}
f_1 &= q_{i_1} \vee q_{i_2} \cdots \vee q_{i_{\beta(1)}}, \\
f_2 &= q_{j_1} \vee q_{j_2} \cdots \vee q_{j_{\beta(2)}}, \\
&\ \ \vdots \\
f_u &= q_{k_1} \vee q_{k_2} \cdots \vee q_{k_{\beta(u)}}.
\end{aligned}
\qquad (10.1)
$$

Not all these terms in the above expressions are distinct. Let q_1, $q_2, \ldots, q_r$ be all the distinct terms appeared in the above expressions. The number of contacts required in implementing f according to the above expressions (10.1) is

$$L + e_1 + e_2 + \ldots + e_r ,$$

where L is the sum of the numbers of literals in $q_1, q_2, \ldots, q_r$, and e_i is the number of times q_i appeared in the expressions (10.1) for each i = 1, 2, ..., r. To minimize (or maximize) the number of contacts required after minimizing the size of PLA in implementing a switching function, one may formulate the logic minimization

problem into the general cost minimal covering problem.

Formulation of the logic minimization problem into the general cost minimal covering problem (GP) can be done in a manner similar to that into the minimal covering problem (P), except the assignment of a cost for each prime implicant. In this formulation, each prime implicant q_i is assigned a cost ($e_i + \ell_i + WW$) or ($-e_i - \ell_i + WW$) depending on whether the problem is to minimize or to maximize the number of contacts required in implementation, where e_i is the number of output functions implied by q_i, ℓ_i is the number of literals in q_i , and WW is a sufficiently large fixed integer to ensure that the number of terms, i.e., the size of the PLA, in the optimal solution is minimized.

Many other important problems [1, 2, 3, 4, 5, 7, 17, 18, 21] can also be formulated into the general cost minimal covering problem. They can then be solved by the generalized algorithm introduced in this chapter. No comparisons with other existing programs on the computational efficiency have been made. Computational results show that the algorithm introduced in this chapter is efficient in solving problems formulated for the logic minimization problem.

10.1 Generalization Of The Basic Algorithm

This section discusses the generalization of the basic algorithm described in Section 3.3.

It is already known [18] that, with a slight modification of the operation 2, the three operations stated in Section 3.2 can be used to reduce the constraint matrix A for the general cost minimal

covering problem. The operation 2 is modified as in the following operation 2'.

Operation 2'. If $\vec{a}_j$ is dominated by column $\vec{a}_i$ and $c_j \geq c_i$, then column $\vec{a}_j$ can be deleted from the matrix and the variable x_j corresponding to column $\vec{a}_j$ is fixed to 0.

The method used in the algorithm of Section 3.3 for calculating a lower bound ZMIN of a subproblem with partial solution S can also be generalized as follows.

For each free variable x_j, let $g_j = c_j/\ell_j$, where ℓ_j is the number of non-zero elements in column j. Arrange g_j s in an increasing order:

$$g_{j_1} \geq g_{j_2} \geq g_{j_3} \geq \; \cdots \; \geq g_{j_p} \; .$$

Let h be the number of unsatisfied constraints by the current partial solution and r be the greatest integer such that

$$\ell_{j_1} + \ell_{j_2} + \ldots + \ell_{j_r} > h \; .$$

Then ZMIN is calculated by $ZMIN = c_{j_1} + c_{j_2} + \ldots + c_{j_r} + w$, where w is the value of the current partial solution S, i. e., $w = \sum_{k \in S} c_k x_k$.

The efficient way described in Section 4.1 for checking domination relation among columns or rows can still be applied to the general cost minimal covering problem.

10.2 Precluding Of Subproblems

It is easy to see that the general cost minimal covering problem also has "the reducing property" as the minimal covering

problem does in Chapter 5.

"The excluding property" in the case of the general cost minimal covering problem is stated in the following theorem.

<u>Theorem 10.2.1</u> Let E_{ij} be the E-set of $\vec{a}_j$ with respect to $\vec{a}_i$, and let $\vec{x}$ be a feasible solution of the general cost minimal covering problem (GP) with $x_i = 0$, $x_j = 1$ and $x_k = 1$ for $j = 1, 2, \ldots, r$ (other variables are assigned either 0 or 1). If each row in E_{ij} is covered by some of columns $\vec{a}_{k_1}, \vec{a}_{k_2}, \ldots, \vec{a}_{k_r}$, then $\vec{x}'$, which is obtained from $\vec{x}$ by replacing $x_i = 0$, $x_j = 1$ with $x_i = 1$, $x_j = 0$ and the remaining variables unchanged, is also a feasible solution of (GP). Furthermore, if $c_j \geq c_i$, then the objective value of $\vec{x}'$ is greater than or equal to the objective value of $\vec{x}$.

<u>Proof</u> The first part of this theorem is exactly the same as Theorem 5.2.1. If $c_j \geq c_i$, then, from the way $\vec{x}$ obtained, the objective value of $\vec{x}$ is greater than or equal to that of $\vec{x}$.

Q.E.D.

From the above theorem, it is easy to see that, among all the subproblems generated at the same time in step M3.2 of the basic algorithm, if the one corresponding to the variable with the smallest cost is enumerated first, then the procedure discussed in Section 5.3 for precluding subproblems is still applicable to the case of the general cost minimal covering problem. Based on the above observation, the criterion used in step M3.1 for selecting a row is modified as follows:

<u>3.1.1</u> For each row in the constraint matrix, find the

smallest cost among all the costs of the columns covered by this row. The number of non-zero elements covered by the column corresponding to this smallest cost will be referred to as "the choosing weight" for this row.

3.1.2 Select the row with the greatest choosing weight among all remaining rows in the constraint matrix. ■

This selection criterion is to find a feasible solution for the problem at the earlist possible iteration under the rule that the subproblem corresponding to the variable with the smallest cost is enumerated first among all the subproblems generated at the same time in step M3.2.

According to the modification discussed in the last and this sections, a generalized algorithm for the general cost minimal covering problem was developed [32]. Some problems were tested by this algorithm. These problems were formulated for the logic minization problem or constructed by the author. By the program for this algorithm, coded in FORTRAN, problems were tested on the CDC Cyber 175 computer. Computational results are shown in Table 10.2.1. Computational results of solving the same problems by using the generalized basic algorithm with no "reducing property" or "excluding property" are also given in this table. Figures in this table are explained in Table 5.4.1.

From this table, one can see that the use of "the reducing property" and "the excluding property" in this generalized algorithm does help in speeding up the enumeration in solving problems. The computational improvement is about 30 % on average.

PROB NO	PROB. SIZE				USING NEW PROPERTIES			WITHOUT USING NEW PROPERTIES		
	m	n	m'	n'	NO. OF ITER.	NO. OF BKTRK	TIME IN SEC	NO. OF ITER.	NO. OF BKTRK	TIME IN SEC
1	60	6o	43	50	549	249	1.39	850	330	1.94
2	60	80	52	76	12410	5587	39.39	16882	6421	47.36
3	55	44	45	43	242	146	0.93	288	153	0.98
4	112	79	83	73	6155	3354	35.24	7920	3636	39.87
5	114	83	89	77	15732	10092	118.87	23178	10599	150.62
6	166	156	87	94	636	319	3.87	966	366	5.09

Table 10.2.1

Comparison of some computational results on two cases - with and without using new properties in solving the general cost minimal covering problem.

Problems 1 and 2 were randomly generated by the author. Other problems were formulated for the logic minimization problem. The cost assigned for each prime implicant is the number of literals in it in these four problems.

From Table 10.2.1, it is easy to see that about 30 % of computation time is saved through the implementation of the new procedures mentioned in this section for precluding subproblems.

These four logic minization problems were further formulated

as the minimal covering problems and were solved by the algorithm outlined in Section 5.3. The solutions obtained and the time spent in these two approaches are compared in Table 10.2.2. The column under "NO. OF VAR." shows the number of variables of the switching function whose expression is to be minimized. The columns under "m", "n", and "TIME IN SEC" are explained in Table 5.4.1. The column under "NO. OF Terms" shows the smallest number of products which may epress the given switching function. The column under "NO OF LITR." shows the number of literals in the set of terms found by each algorithm.

NO. OF VAR.	TABLE SIZE		FORMULATED AS (P)			FORMULATED AS (GP)		
	m	n	TIME IN SEC	NO. OF TERMS	NO. OF LITR.	TIME IN SEC	NO. OF TERMS	NO. OF LITR.
6	55	44	0.2	12	40	0.93	12	38
7	112	79	6.56	18	67	35.24	18	66
7	114	83	25.10	15	53	118.87	15	53
8	166	156	0.58	45	254	3.87	45	254

Table 10.2.2

Comparison of solutions of the logic minimization problem obtained by two different approaches - formulated as the minimal covering problem or the general cost minimal covering problem.

From these results, the differences in the running time of this two algorithms are very large while the differences in the numbers of literals in the two solutions obtained by these two algorithms are very small. Unless the minimization or the maximization of the number of literals after minimizing the number of terms is very

important in implementing a switching function, solving the logic minimization problem by the minimal covering problem is preferrable in terms of computation time.

10.3 The Symmetric Property Of The General Cost Minimal Covering Problem

This section discusses the symmetric property of the general cost minimal covering problem.

A permutation η on $\{x_1, x_2, \ldots, x_n\}$ is said to be a symmetric permutation of the general cost minimal covering problem (GP) if the following conditions are satisfied:

(1) $(\eta(x_i), \eta(x_2), \ldots, \eta(x_n))$ is a feasible solution of (GP) whenever $(x_1, x_2, \ldots, x_n)$ is a feasible solution of (GP).

(2) $c_i = c_j$ if $x_i = \eta(x_j)$.

From the above definition, it is easy to see that the objective values for both feasible solutions $(x_1, x_2, \ldots, x_n)$ and $(\eta(x_1), \eta(x_2), \ldots, \eta(x_n))$ are the same. It is also easy to see that the symmetric permutation defined for the minimal covering problem (P) is a special case of the above definition.

In the following, let us see an example of a symmetric permutation of the general cost minimal covering problem.

Let λ be a symmetric permutation of a switching function f. It is known in Section 7.2 that λ defines a permutation on the set of all prime implicants $\{q_1, q_2, \ldots, q_n\}$ of f as $\lambda(q_i) = \lambda(z_1).\lambda(z_2)\ldots.\lambda(z_k)$ if $q_i = z_1.z_2\ldots.z_k$, where $z_j = y_j$ or $\bar{y}_j$, for $j = 1, 2, \ldots, k$. From this, we see that the numbers of literals in q_i and $\lambda(q_i)$ are the same for each prime implicant q_i

of f.

Since λ is a symmetric permutation of f, if q_i is an implicant of some output function f_j of f, then $\lambda(q_i)$ is also an implicant of the output function f_j, by Lemma 7.2.2. Conversely, if $\lambda(q_i)$ is an implicant of some output function f_j of f, then $\lambda^{-1}(\ (q_i)) = q_i$ is also an implicant of f_j, since λ^{-1}, the inverse of , is also a symmetric permutation of f. Thus the numbers of output functions implied by q_i and $\lambda(q_i)$ are the same for each prime implicant q_i of f.

Let the problem (GP) be the general cost minimal covering problem formulated for the logic minimization problem of f (either minimizing or maximizing the number of contacts required in implementation after the size of PLA is minimized), and let $\tilde{\lambda}$ be a permutation of this problem defined as follows:

$$\tilde{\lambda}(x_j) = x_i \quad \text{if and only if} \quad \lambda(q_i) = q_j \ . \quad (10.3.1)$$

Since the numbers of literals in q_i and $\lambda(q_j)$ are the same and the numbers of output functions implied by q_i and $\lambda(q_i)$ are the same for each prime implicant q_i of f, the cost c_i of variable x_i and the cost c_j of the variable x_j are the same for every pair of variables x_i and x_j such that $\tilde{\lambda}(x_i) = x_j$. From the definition (10.3.1) of , $(\tilde{\lambda}(x_1), \tilde{\lambda}(x_2), \ldots, \tilde{\lambda}(x_n))$ is a feasible solution of the program (GP) whenever $(x_1, x_2, \ldots, x_n)$ is a feasible solution of (GP), by Theorem 7.2.4. Thus $\tilde{\lambda}$ defined by (10.3.1) is a symmetric permutation of the general cost minimal covering problem (GP) formulated for the logic minimization problem of f if λ is a

symmetric permutation of f.

Now let us return to the consideration of the general cost minimal covering problem (GP). By the same argument, Theorem 7.1.1 is also true in the case of the general cost minimal covering problem. A necessary and sufficient condition for a permutation to be a symmetric permutation of the general cost minimal covering problem (GP) is modified in the following Theorem 10.3.1. This modification is due to the fact that the condition (2) in this theorem, which a symmetric permutation must satisfy in the case of the general cost minimal covering problem, is always true in the case of the minimal covering problem. The proof of this theorem is exactly the same as the proof of Theorem 7.4.1 and is not repeated here.

Theorem 10.3.1 A permutation η on $\{ x_1, x_2, \ldots, x_n \}$ is a symmetric permutation of the general cost minimal covering problem (GP) if and only if (1) each row of A dominates some row of A and (2) $c_i = c_j$ for every x_i, x_j such that $\eta(x_i) = x_j$.

Following the same discussion, all the theorems in Section 7.4 through 7.7 for the case of the minimal covering problem (P) are also true in the case of the general cost minimal covering problem (GP). So, in solving the general cost minimal covering problem by the implicit enumeration method, the symmetric property of the problem can be utilized in the same manner as it is utilized in the case of the minimal covering problem to speed up the computation if there are some symmetric permutations of the problem.

No procedure is yet implemented for the utilization of symmetric property in the case of the general cost minimal covering

problem. From the experience with the minimal covering problem, a great improvement in the computational efficiency is expected if the procedure for utilizing the symmetric property of the general cost minimal covering problem is implemented.

10.4 Heuristic Approach For The Large-scale General Cost Minimal Covering Problem

The idea in the heuristic algorithm in Chapter 6 for the large-scale minimal covering problem is applied to develop an heuristic algorithm for the general cost minimal covering problem in this section.

Similar to the heuristic algorithm for the minimal covering problem, this heuristic algorithm also decomposes large-scale subproblems into small-scale subproblems and heuristically solves small-scale subproblems by finding a feasible solution for each of them. The decomposition of large-scale subproblems is done in the same way as it is done in the case of the generalized algorithm in Section 10.1 and 10.2. The small-scale subproblem is solved by the following heuristic procedure.

Procedure HG (Heuristic for the General cost minimal covering problem):

HG1 Choose some column $\vec{a}_{j_o}$ by the criterion which will be described later.

HG2 Delete all rows covered by the column $\vec{a}_{j_o}$ from the constraint matrix.

HG3 Reduce the constraint matrix as much as possible, using the three reduction operations stated in Section 10.1. If the constraint matrix is null, then a feasible

solution is found. Otherwise the algorithm's control goes to step HG1. ■

The criterion used in step HG1 for choosing a column $\vec{a}_{j_o}$ consists of the following steps:

<u>HG1.1</u> For each remaining column $\vec{a}_j$ calculate the "cost per row",

$$w_j = c_j / \ell_j ,$$

where c_j is the cost of the variable x_j and ℓ_j is the number of non-zero elements covered by column $\vec{a}_j$.

<u>HG1.2</u> Choose the column $\vec{a}_{j_o}$ such that w_{j_o} is the smallest. If there is a tie, choose the one with the smallest column index. ■

This heuristic algorithm is modified from the generalized algorithm in Sections 10.1 and 10.2 for the general cost minimal covering problem in the same manner as the heuristic algorithm in Chapter 6 for the minimal covering problem is modified from the algorithm in Chapter 5 for the minimal covering problem. It has the same characteristic as the heuristic algorithm for the minimal covering problem does: if the level limit specified for a problem is sufficently large not to be reached in solving this problem, then the best solution obtained is still an optimal solution of this problem.

Three general cost minimal covering problems randomly generated by the author were tested by this heuristic algorithm. By the program for this heuristic algorithm coded in FORTRAN language,

results are obtained by solving problems on the CDC Cyber 175 computer. Computational results are shown in Table 10.4.1. Values of m, n, m' and n' are explained in Table 5.4.1. The columns under "LEVEL LIMIT", "TIME IN SEC", and "VAL" are the same as those in Table 6.2.1. The column under "NO. OF ITER" shows the number of times the algorithm went through the step M1 of the algorithm under the specified limit shown in the column under "LEVEL LIMIT". "-" in the Table shows that no test was made in that case. "∞" in the column under "LEVEL LIMIT" means that no level limit was specified in the test and the best value obtained in this test was the optimal value of the problem.

From this table one can see that reasonably good solutions can be obtained in a reasonable amount of computation time by specifying the level limit equal to 6 in solving these three general cost minimal covering problems. One can also see that the optimal solutions of these three problems can all be obtained if the level limit specified is 8. From this observation, this heuristic algorithm can be very useful in solving large-scale general cost minimal covering problem if an appropriate level limit is specified.

PROB. NO.	1				2				3			
PROB. SIZE	m	n	m'	n'	m	n	m'	n'	m	n	m'	n'
	60	60	43	52	60	80	52	76	60	60	59	57

PROB. NO.	1			2			3		
LEVEL LIMIT	TIME IN SEC.	NO. OF ITER	VAL	TIME IN SEC.	NO. OF ITER	VAL	TIME IN SEC.	NO. OF ITER	VAL
2	0.07	3	95	0.10	3	208	0.07	3	84
4	0.35	33	90	0.23	81	195	0.57	45	83
6	1.13	186	89	9.29	839	193	3.28	361	81
8	-	-	-	27.06	3790	192	10.10	1672	79
∞	1.42	522	89	39.39	12410	192	15.78	5753	79

Table 10.4.1

Some computational results of the heuristic algorithm for the general cost minimal covering problem; results are obtained by running program on the CDC Cyber 175.

11. CONCLUSION

Efficient implicit enumeration algorithms for the minimal covering problem are presented in this thesis. These algorithms are developed mainly for minimizing the logic expression of the switching function. They are extensions of the Quine-McCluskey method described in [6] for solving the prime implicant table.

The most powerful procedure of the Quine-McCluskey method in solving a logic minimization problem is the repeated use of the problem reduction. An effective procedure for reducing the computation time in the problem reduction is devised in Chapter 4.

"The excluding property" of the minimal covering problem in Chapter 5 is introduced to speed up the enumeration in solving the minimal covering problem. Another property, "the reducing property" of the minimal covering problem is also introduced in Chapter 5, even though the conditions of this property are rarely satisfied by partial solutions in solving actual problems. Computational improvement is about 30 % through the implementation of the procedure described in Section 5.3 for only implementing "the excluding property".

The heuristic algorithm in Chapter 6 is an extension of the algorithm in Chapter 5. Solutions examined by this heuristic algorithm are evenly distributed in the decomposition tree, which represents the decomposition of the given problem into subproblems in the case when this problem is solved by the implicit enumeration algorithm of Chapter

5. This heuristic algorithm is a practical algorithm for the large -scale minimal covering problem.

The symmetric property of the minimal covering problem is introduced in this thesis. This property and its utilization in the implicit enumeration are extensively explored in Chapter 7. The relation between the symmetric property of the switching function and the symmetric property of the minimal covering problem formulated for the logic minimization problem is also discussed in Chapter 7. Utilizing the symmetric property in solving a symmetric minimal covering problem by the implicit enumeration method, the computational improvement is more than ten times for some problems. The computationally difficult minimal covering problems in [15], [24] are symmetric. Minimal covering problems formulated for minimizing the logic expression of symmetric or partially symmetric switching functions are also symmetric.

In Chapter 8 more properties of the minimal covering problem which may be used to speed up the implicit enumeration in solving the minimal covering problem are discussed, through it needs further exploration to effectively utilize these properties in solving the minimal covering problem.

The concept of an upper bound on the value of a group of variables is introduced in Chapter 9. If the constraint matrix of the given minimal covering problem has the partition structure shown in (9.1), then the variables of this problem can be grouped into groups and an upper bound on the value of each group can be found. These upper bounds can be checked in the enumeration procedure to speed up

the implicit enumeration in solving problems. Computational improvement is about 30 % through the checking of an upper bound for each group of variable in the enumeration procedure.

The implicit enumeration algorithm and its extension to the heuristic algorithm discussed in the previous chapters for the minimal covering problem are generalized in Chapter 10 for the general cost minimal covering problem. This generalization is mainly for solving the problem of minimizing the size of PLA required as the first criterion and minizing or maximizing the number of contacts as the secondary criterion in implementing a switching function by PLA. General cost minimal covering problems formulated for other problems [1, 2, 3, 4, 5, 7, 17, 18, 21] can also be solved by the generalized algorithm or by the generalized heuristic algorithm in Chapter 10.

The basic structure of this generalized algorithm is different from the algorithm in [12] in that the value obtained from the relaxed linear programming problem of each subproblem is used as a lower bound on the value of that subproblem, while a very simple procedure is used to estimate the lower bound of each subproblem in this generalized algorithm. "The reducing property" and "the excluding property" are further incorporated in this generalized algorithm to speed up the enumeration. Since the algorithm in [12] is also an implicit enumeration algorithm, "the generalized excluding property" may also be incorporated into that algorithm to improve its computational efficiency. (The algorithm of [12] contains a kind of "the reducing property".) The utilization of the symmetric

property of the given problem discussed in Chapter 10 may also be applied to that algorithm. Computational improvement in solving symmetric problems is expected if the procedure discussed in Section 10.3 for utilizing the symmetric property of the given problem is incorporated in both algorithms.

No computational comparison of both algorithms has been made. Since the algorithm in [12] uses linear programming to find the lower bound of the value of each subproblem, its efficiency completely depends on that of linear programming method used in it. This algorithm may have difficulty in solving problems with symmetric properties such as the two problems reported in [24] or the problem IBM 9 reported in [15], since solving these kind of problems by the implicit enumeration method usually requires a large number of iterations.

The heuristic algorithm described in Section 10.4 is useful in solving large-scale general cost minimal covering problems. In using this heuristic algorithm, if one specifies a value no greater than 10 as the level limit, then one usually can obtain a reasonably good solution for a large-scale general cost minimal covering problem in a reasonable amount of computation time.

The programs developed based on the algorithms discussed in Chapters 5, 6, 7, 9 and 10 are available in [32]. These programs are further incorporated into the ILLOD-MINSUM system [33] for the automated design of two-level AND/OR optimal networks.

REFERENCES

1. Arabyre, J. P., J. Fearnley, F. C. Steiger, and W. Teather, "The Crew Scheduling Problem: A Survey," Transp. Sci. 3, 140-163 (1969).

2. Bellmore, M., H. J. Greenberg, and J. J. Jarvis, "Multi-Commodity Disconnecting Sets," Management Science 16, B427-433 (1970).

3. Day, R. H., "On Optimal Extracting From a Multiple File Data Storage System: An Application of Integer Programming," Operations Research 13, 482-494 (1965).

4. Garfinkel, R. S., and G. L. Nemhauser, "Optimal Political Districting by Implicit Enumeration Techniques," Management Science 16, B495-508 (1970).

5. Balinski, M.L., and R. Quandt, "On an Integer Program for a Delivery Program," Operations Research 12, 300-304 (1964).

6. McClusky, E. J., "Introduction to the Theory of Switching Circuits," McGraw-Hill (1965).

7. Ibaraki, T. and S. Muroga, "Synthesis of Network with a Minimum Number of Negative Gates," IEEE Transaction on Computer, Vol. C-20, No. 1, Jan. 1971 pp. 49-58.

8. Cobham, A., R. Fridshal, J. H. North, "A Statistical Study of the Minimization of Boolean Functions Using Integer Programming," IBM Research Report, RC-756 (1962).

9. Balas, E., "An Additive Algorithm for Solving Linear Programs with 0-1 Variables,"Operations Research 13, 517-546 (1965).

10. Geoffrion, A. M., "An Improved Implicit Enumeration Approach to Integer Programming," Operations Research 17, 437-454 (1969).

11. Ibaraki, T., T. K. Liu, C. R. Baugh, and S. Muroga, "An Implicit Enumeration Program for Zero-One Integer Programming," International Journal of Computer and Information Science, Vol. 1, No. 1, March 1972.

12. Lemke, C. E., H. M. Salkin and K. Spielberg, "Set Covering By Single-Branch Enumeration with Linear Programming Subproblems," Operations Research 19, 998-1022 (1971).

13. Bellmore, M., and H. D. Ratliff, "Set Covering and Involutory Bases," Management Science, Vol. 18, No. 3, November, 1971, pp. 194-206.

14. Cobham, A., R. Fridshal, and J. H. North, "An Application of Linear Programming to the Minimization of Boolean Functions," IBM Research Report RC-472, 1961.

15. Haldi, j., "25 Integer Programming Test Problems," Working Paper No. 43, Graduate School of Business, Stanford University, December 1964.

16. Kolner, T. N., "Some Highlights of a Scheduling Matrix Generator System," United Airlines, Presented at the Sixth AGIFORS Symposium, Sept. 1966.

17. Wagner, W.H., "An Application of Integer Programming to Legislative Redistricting," Presented at the 34th National Meeting of ORSA, November, 1968.

18. Balinski, M. L., "Integer Programming: Methods, Uses, Computation," Management Science, Vol. 12 (1965), pp. 253-313.

19. Shapiro, J. F., "Group Theoretic Algorithms for the Integer Programming Problem-II: Extension to a General Algorithm," Operations Research 16, 928-947.

20. Trauth, C. A. and R. E. Woolsey (1969), "Integer Linear Programming: A Study in Computational Efficiency," Man. Sci. 15, 481-493.

21. Ibaraki, ., "Gate-Interconnection Minimization of Switching Networks Using Negative Gates," IEEE Transaction on Computers, June 1971, pp. 698-706.

22. Bowman, R. M., and E. S. McVey, "A Method for the Fast Approximate Solutions of Large Prime Implicant Charts," IEEE Transaction on Computers, Feb. 1970, pp. 169-173.

23. Roth, R., "Computer Solution to Minimum-Covering Problems," Operations Research 17, 1969, pp. 455-465.

24. Fulkerson, D. R., G. L. Nemhauser and Trotter Jr., "Two Computational Difficult Set Covering Problems That Arise in Computing the 1-width of Steiner Triple Systems," Mathematical Programming Study 2 (1974), 72-81, North-Holland Publishing Company.

25. Trotter, L. E. Jr. and C. M. Shetty, "An Algorithm for the Bounded Variable Integer Programming Problem," Journal of the Association for Computing Machinery, Vol. 21, No. 3, July 1974, 505-513.

26. Standard EDP Report, AUERBACH INFO, INC., 1972.

27. Computer Characteristic Quarterly, 1968.

28. Lawler, E. L., "Covering Problems: Duality Relations and A New Method of Solution," SIAM J. Appl. Math. 14, 1115-1132, 1966.

29. Lemke, C. and K. Spielberg, "Direct Search 0-1 and Mixed Integer Programming," Operations Research 15 (1967), pp. 892-914.

30. Muroga, S., "Logic Design and Switching Theory," to be published in 1979 by John Wiley.

31. Cutler, R. B., "MINSUM: A Library of Subroutines for Finding Irredundant Disjunctive Forms or Minimal Sums for Switching Functions - Subroutine Descriptions," to appear.

32. Young, M. H., "Program Manual of Programs for Minimal Covering Problems: ILLOD-MINIC-B, ILLOD-MINIC-BP, ILLOD-MINIC-BS, ILLOD-MINIC-BA, ILLOD-MINIC-BG," Report No. UIUCDCS-R-78-924, Dept. of Computer Science, University of Illinois, 1978.

33. Young, M. H. and R. B. Cutler, "Program Manual for the Programs ILLOD-MINSUM-CBS, ILLOD-MINSUM-CBSA, ILLOD-MINSUM-CBG, ILLOD-MINSUM-CBGM, To Derive Minimal Sums Or Irredundant Disjunctive Forms for Switching Functions," Report No. UIUCDCS-R-78-926, Department of Computer Science, University of Illinois, 1978.

34. Carmichael, R. D., "Introduction to the Theory of Group of finite order," p. 8, Dover Publications, Inc. 1956.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-79-966	2.	3. Recipient's Accession No.
4. Title and Subtitle THE MINIMAL COVERING PROGLEM AND AUTOMATED DESIGN OF TWO-LEVEL AND/OR OPTIMAL NETWORKS			5. Report Date March 1979
			6.
7. Author(s) Ming Huei Young			8. Performing Organization Rept. No. UIUCDCS-R-79-966
9. Performing Organization Name and Address University of Illinois at Urbana-Champaign Department of Computer Science Urbana, Illinois 61801			10. Project/Task/Work Unit No.
			11. Contract/Grant No. NSF MCS77-09744
12. Sponsoring Organization Name and Address National Science Foundation Washington, DC			13. Type of Report & Period Covered Ph.D. Thesis
			14.

15. Supplementary Notes

16. Abstracts

Efficient implicit enumeration algorithms for the minimal covering problem are presented in this thesis. These algorithms are developed mainly for minimizing the logic expression of the switching function. They are extensions of the Quine-McCluskey method.

"The reducing property" and the "excluding property" of the minimal covering problem are introduced to speed up the enumeration in solving problems.

Symmetric property of the minimal covering problem is extensively explored. Procedures for utilizing this property in the implicit enumeration algorithm are developed based on the theory of finite permutation group.

The concept of an upper bound on the value of a group and of variable is also introduced in this thesis.

Programs developed based on these algorithms are incorporated into a system for the automated design of two-level AND/OR optimal networks.

17. Key Words and Document Analysis. 17a. Descriptors

Minimal covering problem,
AND gate,
OR gate,
Logic design,
Optimal network,
Symmetric switching function,
0-1 variable programming problem,
Permutation.

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 192
Unlimited	20. Security Class (This Page UNCLASSIFIED	22. Price

FORM NTIS-35 (10-70) USCOMM-DC 40329-P71